TOWARDS ELECTRONIC STRUCTURE CALCULATIONS AT THE EXASCALE

A Thesis
Presented to
The Academic Faculty

by

Phanisri Pradeep Pratapa

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Civil and Environmental Engineering

Georgia Institute of Technology
August 2016

TOWARDS ELECTRONIC STRUCTURE CALCULATIONS AT THE EXASCALE

Approved by:

Professor Phanish Suryanarayana,
Advisor
School of Civil and Environmental Engineering
Georgia Institute of Technology

Professor Glaucio H. Paulino
School of Civil and Environmental Engineering
Georgia Institute of Technology

Professor Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Arash Yavari
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. John E. Pask
Physics Division
Lawrence Livermore National Laboratory

Date Approved: 7 July 2016

ACKNOWLEDGEMENTS

I sincerely thank my advisor Prof. Phanish Suryanarayana, without whom I would not have been able to contribute through my doctoral research. Prof. Suryanarayana, had helped me overcome many obstacles in my work and provided directions that led to fruitful results. I thank him for providing me an opportunity to pursue doctoral research and patiently helping me learn advanced technical concepts during the course of my PhD.

I gratefully acknowledge the funding grants from National Science Foundation and Lawrence Livermore National Laboratory that supported my graduate study. I thank Dr. John Pask, Prof. Edmond Chow, Prof. Glaucio Paulino and Prof. Arash Yavari for serving on my thesis committee. Most of my work has been in collaboration with Dr. Pask and his guidance and support to our research has been invaluable.

I thank all my friends here at Georgia Tech with whom I had spent many fun filled and memorable weekends. I thank Prasanth Alapati, Bala Chandra Suri, Janani Venugopalan, Lakshmi Priya Subramanian, Trilochan Rambhatla, Ajinkya Lokhande and Qimen Xu (Jim). I thank Swarnava Ghosh, my labmate and friend, for being helpful and co-operative.

I thank my parents for being supportive of my decision to pursue doctoral studies. I also thank my sister Seshasree Harika, brother-in-law Kiran Vemu and my nephew Asrithsai Pradyumna for spending quality time with me. I thank my parents-in-law for the continuous encouragement, counselling and motivation they have provided during my study. I thank Sriram Malladi for many friendly conversations and discussions.

Finally, I would like to mention my wife Satya Sarvani, for always being there

with me in the ups and downs of a PhD journey. Given that she has also been a doctoral student during my PhD, the help and support I received from her is beyond words.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Development of new materials needs better understanding of the behavior of materials at nanoscale which involves accurate simulation of atomic and electronic interactions. Electronic structure is especially important when the atomic interactions involve breaking or formation of chemical bonds. When such interactions are present, first principles based ab-initio electronic structure calculations of atoms, which do not involve any empirical potentials, would be a suitable choice to study the behavior of materials at nanoscale. Such simulations involving many thousands of atoms are intractable by current software (especially for metals) due to their cubic scaling with respect to the system size. In this dissertation, the cubic scaling bottleneck is overcome by developing a linear scaling method amenable to massive parallelization.

A linear scaling Density Functional Theory (DFT) framework has been developed using Clenshaw-Curtis Spectral Quadrature (SQ) method and implemented on massively parallel computers to simulate the electronic structure of hundreds of thousands of atoms. Finite difference representation has been employed in order to exploit the locality of electronic interactions in real space, enable systematic convergence and facilitate large-scale parallel implementation. In combination with linear scaling electrostatics, the electron density, energy and atomic forces can be calculated with effort that scales linearly with the number of atoms for both insulating and metallic systems. This method allows computation of the Γ-point and infinite-cell calculations without resorting to Brillouin zone integration or large supercells.

The method is validated and systematic convergence of energy and forces to the exact diagonalization result is demonstrated. Convergence with respect to mesh size to established cubic scaling planewave results has also been shown. The efficiency

and suitability of the method for high temperature calculations is also discussed. Energy and forces for systems with many thousands of atoms have been computed. The parallel scaling of the method to more than hundred thousand processors has been studied. The extreme parallelizability demonstrated by the method promises the potential to make use of the next generation exascale computer architectures for scientific simulations. In the spirit of massive parallelizability and efficiency, new extrapolation techniques have been developed to accelerate the convergence of fixed point iterations. These techniques when applied to basic iterative methods give rise to efficient solvers for linear systems of equations. Robust and efficient performance of these methods is demonstrated in acceleration of the non-linear fixed point iteration that is used to solve the electronic structure problem.

The SQ method enables simulation of very large systems of both metals and insulators under a unified framework, at high temperatures. It also enables performing ab-initio molecular dynamics simulations at high temperatures which is impractical using cubic-scaling codes. This method also provides the basis on which an accurate simulation of the mechanics of materials at nanoscale can be performed in multi-scale modeling studies using coarse graining techniques.

CHAPTER I

INTRODUCTION

1.1 Motivation

Modeling of materials and their mechanical behavior is crucial in characterizing and designing new materials with exotic properties. Defects play a key role in determining the macroscopic properties of solids [82]. Predictive modeling of materials requires accurate depiction of defects which involves chemical interactions between atoms. Although, atomistic modeling techniques that use empirical potentials exist [109, 87], they do not accurately capture the electronic interactions during bond formation or bond breaking. Hence first principles based ab-initio electronic structure calculations of atoms, which do not involve any empirical potentials, would be a suitable choice to study the behavior of materials at nanoscale. This entails simulating atomic systems that has much more than thousands of atoms which is intractable by present day electronic structure software available. The primary reason for this is that most of the codes scale as $\mathcal{O}(N^3)$ and/or are not massively parallelizable and can practically simulate only a few hundreds to thousands of atoms. Here, N refers to the number of atoms in the system. The goal of this work is to present an $\mathcal{O}(N)$ method for electronic structure calculation and develop a parallel framework that can potentially be run on exascale computing architectures which will enable the simulation of hundreds of thousands of atoms. In this work we use Density Functional Theory [49, 54] which is one of the most widely used ab-initio frameworks for electronic structure calculations.

1.2 Organization of the thesis

This dissertation is organized as follows. In this chapter (Ch. 1), we discuss the background on Density Functional Theory (DFT) using orbital formulation, Bloch's theorem and the cubic scaling behavior of the formulation.

In Chapter 2, we discuss the linear scaling DFT and present the Clenshaw-Curtis spectral quadrature approach to solve the problem. We present the expressions for energy and forces and present an outline of the entire DFT problem to solve for the electronic structure of atoms.

In Chapter 3, we present improved extrapolation techniques based on Anderson extrapolation which is an acceleration technique for fixed point iterations. We show that these techniques give rise to efficient linear solvers when applied to Jacobi iteration. Specifically, we develop the restarted Pulay Jacobi and the Alternating Anderson Jacobi methods as solvers for linear system of equations.

In Chapter 4, we discuss the numerical and parallel implementation of the $\mathcal{O}(N)$ DFT method. We derive expressions to compute electron density, energy and forces. We also discuss the scalability of the implementation and memory requirements.

In Chapter 5, we first validate the method through a one-dimensional model problem and then demonstrate the accuracy and convergence of the method for Kohn-Sham DFT calculations. We discuss the efficiency of this method for high temperature calculations. We then demonstrate the strong and weak parallel scalability of the method through large-scale simulations.

The literature review for various topics in this thesis is discussed in the respective chapters. Finally, we summarize and conclude in in Chapter 6 with discussion on the scope for future work.

1.3 Density Functional Theory (DFT)

Over the course of the past few decades, Density Functional Theory (DFT) of Hohenberg, Kohn, and Sham [49, 54] has been widely employed for understanding and predicting a wide range of materials properties. Indeed, DFT calculations are based on the first principles of quantum mechanics, which makes them free of any empirical parameters. DFT involves computing the electron density of a system of atoms by solving a fixed-point iteration called Self-Consistent Field (SCF) iteration [28, 63]. In every iteration of SCF we also solve the Poisson's equation to compute the electrostatic potential [76] from the electrons and the nuclei.

Consider an infinite periodic system of atoms in $\mathbb{R}^3$. Let the unit cell of the infinite periodic system, denoted by Ω, contain N atoms with a total of N_Ω valence electrons. The infinite system is obtained by replicating the Ω domain of atoms over all of $\mathbb{R}^3$. Let $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N\}$ denote the set of position vectors of all the nuclei with charges $\{Z_1, Z_2, \ldots, Z_N\}$ respectively. We igonore spin in the following expressions for clarity. Assuming periodic boundary conditions on the orbitals $\psi_i(\mathbf{x})$, the energy of the system can be written as [75, 24, 34, 32]

$$\begin{aligned}\mathcal{E}(\Psi, \mathbf{g}, \mathbf{R}) &= -\sum_{n=1}^{N_{\text{orb}}} g_n \int_\Omega \psi_n^*(\mathbf{x}) \nabla^2 \psi_n(\mathbf{x})\, \mathrm{d}\mathbf{x} + E_{\text{xc}}(\rho) \\ &\quad + E_{\text{H}}(\rho) + E_{\text{ext}}(\rho, \mathbf{R}) + E_{\text{zz}}(\mathbf{R}) + K(\Psi, \mathbf{R})\,, \end{aligned} \tag{1}$$

where $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_{\text{orb}}}\}$ is the vector of orbitals, $\mathbf{g} = \{g_1, g_2, \ldots, g_{N_{\text{orb}}}\}$ is the vector of orbital occupations, and the electron density

$$\rho(\mathbf{x}) = 2 \sum_{n=1}^{N_{\text{orb}}} g_n |\psi_n(\mathbf{x})|^2. \tag{2}$$

Here, N_{orb} is the maximum number of states or orbitals required by the system, which is proportional to the number of atoms N. The first term in Eqn. 1 represents the kinetic energy of the non-interacting electrons. The second term, $E_{\text{xc}}(\rho)$, denotes the exchange-correlation energy. Variants of this term include the Local Density

Approximation (LDA) [54, 79] and the Generalized Gradient Approximation (GGA) [61, 80]. The next three terms account for the electrostatics:

$$E_{\mathrm{H}}(\rho) = \frac{1}{2}\int_{\Omega}\int_{\mathbb{R}^3} \frac{\rho(\mathbf{x})\rho(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{x}'\,, \tag{3}$$

$$E_{\mathrm{ext}}(\rho,\mathbf{R}) = \int_{\Omega}\rho(\mathbf{x})V_{\mathrm{ext}}(\mathbf{x},\mathbf{R})\,\mathrm{d}\mathbf{x}\,, \tag{4}$$

$$E_{\mathrm{zz}}(\mathbf{R}) = \frac{1}{2}\sum_{I=1}^{N}\sum_{J\neq I}\frac{Z_I Z_J}{|\mathbf{R}_I-\mathbf{R}_J|}\,. \tag{5}$$

$E_{\mathrm{H}}(\rho)$ is known as the Hartree energy and is the classical electrostatic interaction energy of the electron density, $E_{\mathrm{ext}}(\rho,\mathbf{R})$ is the electrostatic interaction energy between the electron density and nuclear charges, and $E_{\mathrm{zz}}(\mathbf{R})$ is the repulsive energy between the nuclei. Finally,

$$V_{\mathrm{ext}}(\mathbf{x},\mathbf{R}) = \sum_{J} V_{\mathrm{loc}}^{J}(\mathbf{x},\mathbf{R}_J)\,, \tag{6}$$

where $V_{\mathrm{loc}}^{J}(\mathbf{x},\mathbf{R}_J)$ denotes the local ionic component of the pseudopotential [83]. In the above expressions, the summation index J runs over all the atoms in Ω as well as their periodic images in $\mathbb{R}^3$.

The last term in Eqn. 1 represents the contribution from the non-local components of the pseudopotential. In this work, we are interested in incorporating the Troullier-Martins pseudopotential [111] in the Kleinman-Bylander [53] form. In this case,

$$\mathcal{K}(\Psi,\mathbf{R}) = 2\sum_{n=1}^{N_{\mathrm{orb}}} g_n \sum_{J=1}^{N}\sum_{lm}\left\langle \Delta V_{lm}^{J}\right\rangle \left[G_{n,lm}^{J}\right]^2\,, \tag{7}$$

where

$$G_{n,lm}^{J} = \frac{1}{\left\langle \Delta V_{lm}^{J}\right\rangle}\int_{\mathbb{R}^3} u_{lm}^{J}(\mathbf{x},\mathbf{R}_J)\Delta V_{l}^{J}(\mathbf{x},\mathbf{R}_J)\psi_n(\mathbf{x})\,\mathrm{d}\mathbf{x}\,, \tag{8}$$

$$\left\langle \Delta V_{lm}^{J}\right\rangle = \int_{\mathbb{R}^3} u_{lm}^{J}(\mathbf{x},\mathbf{R}_J)\Delta V_{l}^{J}(\mathbf{x},\mathbf{R}_J)u_{lm}^{J}(\mathbf{x},\mathbf{R}_J)\,\mathrm{d}\mathbf{x}. \tag{9}$$

In the above expressions

$$\Delta V_{l}^{J}(\mathbf{x},\mathbf{R}_J) = V_{l}^{J}(\mathbf{x},\mathbf{R}_J) - V_{\mathrm{loc}}^{J}(\mathbf{x},\mathbf{R}_J). \tag{10}$$

$V_l^J(\mathbf{x}, \mathbf{R}_J)$ is the ionic pseudopotential component corresponding to the azimuthal quantum number l and $u_{lm}^J(\mathbf{x}, \mathbf{R}_J)$ represents the pseudo-wavefunction for the valence states of interest, all for a single atom. The superscript J is for the atom number and the subscript m denotes the magnetic quantum number.

At finite temperature [69], the contribution to the free energy resulting from the Fermi-Dirac distribution of fractional occupations of the orbitals is given by [75, 24]

$$\mathcal{S}(\mathbf{g}) = -2\sigma \sum_{n=1}^{N_{\text{orb}}} \left[g_n \log g_n + (1 - g_n) \log(1 - g_n) \right] , \tag{11}$$

where the smearing $\sigma = k_B T$, k_B being the Boltzmann constant and T the temperature. Finite values of smearing are typically utilized for metallic systems and when performing finite temperature calculations.

Variational Problem: For a given position of the nuclei $\mathbf{R}$, the electronic ground-state energy of the system is obtained as the solution to the variational principle [75, 24]

$$\mathcal{E}_0(\mathbf{R}) = \inf_{\Psi, \mathbf{g}} \{ \mathcal{E}(\Psi, \mathbf{g}, \mathbf{R}) - \mathcal{S}(\mathbf{g}) \} \tag{12}$$

subject to the orthonormal constraint on the orbitals

$$\int_{\mathbb{R}^3} \psi_i^*(\mathbf{x}) \psi_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \delta_{ij} , \qquad i, j = 1, 2, \ldots, N_{\text{orb}} \tag{13}$$

and the constraint on the total number of valence electrons

$$2 \sum_{n=1}^{N_{\text{orb}}} g_n = N_\Omega . \tag{14}$$

Nonlinear eigenvalue problem: On taking the first variation in Eqn. 12, the DFT problem for a fixed position of the nuclei takes the form [75, 66]

$$\begin{aligned} \mathcal{H}\psi_n &= \lambda_n \psi_n , \quad \mathcal{H} = -\frac{1}{2}\nabla^2 + V_{\text{eff}}(\rho, \mathbf{R}) + \mathcal{V}_{\text{nl}}(\mathbf{x}, \mathbf{R}) , \quad n = 1, 2, \ldots N_{\text{orb}} \\ \rho(\mathbf{x}) &= 2 \sum_{n=1}^{N_{\text{orb}}} g_n |\psi_n(\mathbf{x})|^2 , \qquad (15) \\ g_n &= g(\lambda_n, \mu) = \frac{1}{1 + \exp(\frac{\lambda_n - \mu}{\sigma})} . \qquad (16) \end{aligned}$$

The Fermi energy μ in the Fermi-Dirac distribution of Eqn. 16 is determined by satisfying the constraint of the total number of electrons given by,

$$N_\Omega = 2\sum_{n=1}^{N_{\text{orb}}} g_n.$$

Further, the effective potential

$$V_{\text{eff}}(\rho, \mathbf{R}) = V_{xc}(\rho) + V_H(\rho) + V_{\text{ext}}(\mathbf{x}, \mathbf{R})\,, \tag{17}$$

where

$$V_{xc}(\rho) = \frac{\delta E_{xc}(\rho)}{\delta \rho}\,, \tag{18}$$

$$V_H(\rho) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} \mathrm{d}\mathbf{x}'. \tag{19}$$

The nonlocal part of the pseudopotential represents an angular momentum dependent operator on the orbitals:

$$\mathcal{V}_{\text{nl}}(\mathbf{x}, \mathbf{R})\psi_n(\mathbf{x}) = \sum_{J=1}^{N} \sum_{lm} G^J_{n,lm} u^J_{lm}(\mathbf{x}, \mathbf{R}_J) \Delta V_l{}^J(\mathbf{x}, \mathbf{R}_J) \tag{20}$$

Above, $V_{xc}(\rho)$, $V_H(\rho)$ and $V_{\text{ext}}(\mathbf{x}, \mathbf{R})$ are referred to as the exchange-correlation, Hartree and external potentials respectively. This problem is typically solved by a fixed point iteration with respect to the electron density, known as the self-consistent field (SCF) method [66]. In each iteration of the SCF method, the electron density is calculated by solving for the eigenfunctions ψ_n corresponding to the lowest N_{orb} eigenvalues λ_n, and then using Eqn. 2.

1.4 Bloch's theorem

The orbital formulation of DFT in the above section corresponds to periodic boundary conditions on the unit cell Ω which is called a Γ-point calculation. For sufficiently large Ω (a few hundred to thousand atoms, depending on physical system), these boundary conditions suffice to obtain the infinite-crystal result, corresponding to repetition

of Ω over all space. However, for less complex systems (e.g., elemental solids or compounds) with fewer atoms in Ω, periodic boundary conditions are not sufficient. In such cases, the standard approach is to employ Bloch boundary conditions [7],

$$\psi_{n\mathbf{k}}(\mathbf{x}+\mathbf{L}) = e^{i\mathbf{k}.\mathbf{L}}\psi_{n\mathbf{k}}(\mathbf{x}) \tag{21}$$

where $\mathbf{L}$ is a Bravais lattice vector, $\mathbf{k}$ is the wave vector and $\mathbf{x} \in \Omega$. The desired infinite-crystal (or infinite-cell) result can be obtained as an integral over the Brillouin zone [66]. However, this requires a separate Kohn-Sham calculation for each Bloch wavevector (k-point) in the integration, which can increase cost substantially. For example, metallic systems with small Ω at high pressure can require thousands of k-points to converge. The alternative in such a case would be a periodic calculation on a cell containing thousands of atoms, which for standard diagonalization based approaches would be prohibitive. Since the computational cost for such methods increases cubically with the number of atoms but only linearly with the number of k-points, computation on the small cell with integration over the Brillouin zone is generally the less costly alternative. It can be noted that for Γ-point calculation, $\mathbf{k} = \mathbf{0}$ in Eqn. 21 and $\psi(\mathbf{x})$ is periodic.

1.5 Cubic scaling bottleneck

The tremendous popularity of DFT stems from its high accuracy to cost ratio when compared to other ab-initio theories. However, the solution of the required Kohn-Sham equations — with Schrödinger type three-dimensional eigenproblem for the orbitals — is still a formidable task. This has severely limited the range of physical systems accessible to such rigorous quantum mechanical investigation. In fact, routine calculations are currently restricted to hundreds of atoms. The bottleneck in nearly all DFT calculations is the solution of the eigenproblem for the orthonormal eigenfunctions. Since the required number of eigenfunctions is proportional to the number of atoms in the system N, the overall computational complexity for DFT

simulations is the highly restrictive $\mathcal{O}(N^3)$ [66, 24]. The need for storage of all the eigenfunctions means that the memory costs scale as $\mathcal{O}(N^2)$. Furthermore, the need for orthogonality among the eigenfunctions results in global communications between the processors, thereby severely hindering parallel scalability. The need for high performance computing is especially acute in the context of *ab initio* molecular dynamics [67, 56], wherein the Kohn-Sham equations must be solved at each time step, thereby requiring up to tens of thousands of force evaluations. In Fig. 1, we compare the proposed linear scaling method in this work (which we call SQDFT) and a widely used diagonalization based code called ABINIT [40]. The calculations correspond to an Aluminum lattice with electronic smearing of 4 eV and the parameters have been chosen to be just sufficient to give chemical accuracy in both the codes. The slope of the line through the points corresponding to SQDFT is one and the slope of the line corresponding to ABINIT is two. In the limit of larger systems, we expect that diagonalization costs dominate and the scaling of ABINIT would be close to three. We note that these results correspond to serial single core calculations and the high prefactor associated with the scaling of SQDFT method can be efficiently mitigated through its ability to scale to massively parallel clusters which makes it competitive even for smaller sized systems. This is validated from the results in Section 5.5.

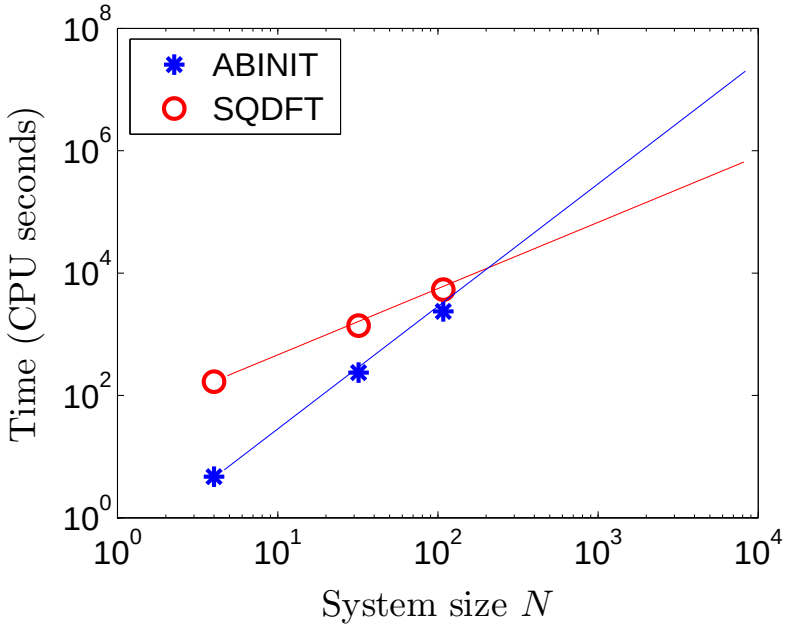


Figure 1: Comparison of a linear scaling method (SQDFT proposed in this work) and a diagonalization based code (ABINIT). The runs were carried out on a single processor.

CHAPTER II

LINEAR SCALING DENSITY FUNCTIONAL THEORY

To overcome the critical $\mathcal{O}(N^3)$ scaling bottleneck, much work has been done in the past two decades to develop solution strategies that scale linearly with the number of atoms, i.e., $\mathcal{O}(N)$ (see, e.g., [37, 13] and references therein). These methods eliminate the computation of the Kohn-Sham orbitals, proceeding instead directly from Hamiltonian to density and total energy without diagonalization. This can be achieved through the density matrix (or density operator) formulation discussed in the following section.

2.1 $\mathcal{O}(N)$ Density Functional Theory

Linear scaling behavior is required in the computation of each and every component of the entire DFT calculation. Electron density and the electrostatics computations are the key steps that need special treatment to enable linear scaling. These two are discussed in the following two sub-sections. The first is addressed through the density matrix formulation and the latter is addressed through a local re-formulation of the electrostatics and by solving Poisson's equation.

2.1.1 Density matrix formulation

We discuss the density matrix formulation for the Γ-point problem. Again, we consider the periodic system of atoms in Ω (Sec. 1.3) with $\mathbf{R}$ denoting the set of position vectors of all the nuclei. We ignore spin in the following expressions for clarity. Using density operator, the system's free energy can be expressed as [3]

$$\mathcal{F}(\mathcal{D}, \mathbf{R}) = 2\mathrm{Tr}\left(-\frac{1}{2}\nabla^2\mathcal{D}\right) + E_{xc}(\rho) + 2\mathrm{Tr}(\mathcal{V}_{\mathrm{nl}}\mathcal{D}) + E_{el}(\rho, \mathbf{R}) - T\mathcal{S}(\mathcal{D})\,, \qquad (22)$$

where Tr(.) denotes the trace, $\mathcal{D}$ is the density operator, $\mathcal{V}_{\text{nl}}$ is the nonlocal pseudopotential operator [83], T is the electronic temperature, and the electron density is given by

$$\rho(\mathbf{x}) = 2\mathcal{D}(\mathbf{x}, \mathbf{x})\,. \tag{23}$$

In Eqn. 22, the first term is the kinetic energy of the non-interacting electrons, the second term is the exchange-correlation energy, the third term corresponds to the nonlocal pseudopotential energy, and the fourth term is the energy due to the electrostatic interactions between the electrons and the nuclei. The electronic entropy contribution to the energy is given by the final term with

$$\mathcal{S}(\mathcal{D}) = -2k_B \text{Tr}\left(\mathcal{D}\log\mathcal{D} + (\mathcal{I}-\mathcal{D})\log(\mathcal{I}-\mathcal{D})\right)\,, \tag{24}$$

where k_B is the Boltzmann constant, and $\mathcal{I}$ is the identity operator. All of the operators in the above equations are defined on $H^1(\Omega)$ which is a space of functions in $L^2(\Omega)$ that have their first derivatives also in $L^2(\Omega)$, with Ω subjected to periodic boundary conditions.

We utilize the Local Density Approximation (LDA) [54] for the exchange-correlation energy:

$$E_{xc}(\rho) = \int_{\mathbb{R}^3} \varepsilon_{xc}(\rho(\mathbf{x}))\rho(\mathbf{x})\,\mathrm{d}\mathbf{x}\,, \tag{25}$$

where $\varepsilon_{xc}(\rho)$ is the sum of the exchange and correlation energy per particle of a uniform electron gas. Using the Kleinman-Bylander form [53] and braket notation, the nonlocal pseudopotential operator acting on a function is expressed as:

$$\begin{aligned}\mathcal{V}_{\text{nl}}\,\psi(\mathbf{x}^{'}) = \sum_J \mathcal{V}_{\text{nl}}^J\,\psi(\mathbf{x}^{'}) &= \sum_J \sum_{lm} \gamma_l^J\,|\chi_{lm}^J\rangle\,\langle\chi_{lm}^J|\psi\rangle \\ &= \sum_J \sum_{lm} \gamma_l^J \chi_{lm}^J(\mathbf{x}^{'}) \int_{\mathbb{R}^3} \chi_{lm}^J(\mathbf{x})\psi(\mathbf{x})\,\mathrm{d}\mathbf{x}\,,\end{aligned} \tag{26}$$

where χ_{lm}^J are the projection functions (local in real space) for the J^{th} atom, with l and m representing the azimuthal and magnetic quantum numbers, respectively. The summation index J runs over all the N atoms in Ω for Γ-point calculation.

The ground-state density operator is the solution to the variational problem [3, 24], for a given configuration of nuclei,

$$\mathcal{D}^* = \left\{ \arg\inf_{\mathcal{D}} \mathcal{F}(\mathcal{D}, \mathbf{R}) \quad \text{s.t.} \quad 2\text{Tr}(\mathcal{D}) = N_\Omega \right\} . \tag{27}$$

where N_Ω is the number of electrons in Ω. The Euler-Lagrange equation for the above problem is a nonlinear fixed-point problem:

$$\mathcal{D} = g(\mathcal{H}, \mu, \sigma) = \left(1 + \exp\left(\frac{\mathcal{H} - \mu\mathcal{I}}{\sigma} \right) \right)^{-1} , \tag{28}$$

where $\sigma = k_B T$ is the smearing, the Fermi energy μ is the Lagrange multiplier used to enforce the constraint on number of electrons, $\mathcal{I}$ is the Identity operator and the Hamiltonian

$$\mathcal{H} = -\frac{1}{2}\nabla^2 + V_{xc} + V_H + V_{\text{ext}} + \mathcal{V}_{\text{nl}} . \tag{29}$$

In the above expression, $V_{xc} = \delta E_{xc}/\delta\rho$ is the exchange-correlation potential, V_H and V_{ext} are the Hartree and external potential given by,

$$V_H(\mathbf{x}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} \mathrm{d}\mathbf{x}' , \tag{30}$$

$$V_{\text{ext}}(\mathbf{x}) = \sum_J V_{\text{loc}}^J(\mathbf{x}, \mathbf{R}_J) . \tag{31}$$

where $V_{\text{loc}}^J(\mathbf{x}, \mathbf{R}_J)$ denotes the local ionic component of the pseudopotential due to the ionic cores located at $\mathbf{R}_J$. The summation index J runs over all of the atoms in $\mathbb{R}^3$.

The resulting density matrix has exponential decay in real-space for insulators as well as metallic systems at finite temperature [36, 10], a key property exploited by $\mathcal{O}(N)$ electronic structure methods. The fixed-point problem of Eqn. 28 is solved using a self-consistent field (SCF) iteration.

2.1.2 Local formulation for the electrostatics

The electrostatic contribution to the energy E_{el} in Eqn. 22 is the sum of E_H, E_{ext} and E_{zz} (given by Eqns. 3, 4 and 5 respectively), involves integrals and summations

over all of $\mathbb{R}^3$ due to long-range interactions. Hence, their direct evaluation would be infeasible. Typically the electrostatics computation is carried out by using Ewald's summation techniques [29] which involves comptuing Fast Fourier Transforms (FFT). Although such techniques could compute electrostatics in a near linear scaling fashion, they are limited by their parallel scalability due to global communications.

In this work, we are interested in computing the electrostatics using a real space framework that scales as $\mathcal{O}(N)$ and is massively parallelizable. We introduce the electrostatic potential $\phi(\mathbf{x}, \mathbf{R}) = V_H + V_{\text{ext}}$ as the solution to the Poisson's equation [77, 106]

$$-\frac{1}{4\pi}\nabla^2\phi(\mathbf{x}, \mathbf{R}) = \rho(\mathbf{x}) + b(\mathbf{x}, \mathbf{R}) \tag{32}$$

where $b(\mathbf{x}, \mathbf{R}) = \sum_J b_J(\mathbf{x}, \mathbf{R}_J)$ denotes the total charge density of the nuclei, with $b_J(\mathbf{x}, \mathbf{R}_J)$ representing the regularized charge density of the J^{th} nucleus i.e.

$$b_J(\mathbf{x}, \mathbf{R}_J) = -\frac{1}{4\pi}\nabla^2 V_{\text{loc}}^J(\mathbf{x}, \mathbf{R}_J). \tag{33}$$

Since $V_{\text{loc}}^J(\mathbf{x}, \mathbf{R}_J)$ replicates the Coulomb potential outside some small core cutoff radius, $b_J(\mathbf{x}, \mathbf{R}_J)$ are localized in space and therefore can be calculated in $\mathcal{O}(N)$ time. Thereafter, the electrostatic energy may be rewritten as [105]

$$\begin{aligned} E_{el}(\rho, \mathbf{R}) = E_{\text{H}} + E_{\text{ext}} + E_{\text{zz}} \;\; = \;\; & \int_\Omega (\rho(\mathbf{x}) + b(\mathbf{x}, \mathbf{R}))\phi(\mathbf{x}, \mathbf{R})\, \mathrm{d}\mathbf{x} \\ & -\frac{1}{2}\sum_J \int_\Omega b_J(\mathbf{x}, \mathbf{R}_J) V_{\text{loc}}^J(\mathbf{x}, \mathbf{R}_J)\, \mathrm{d}\mathbf{x} + \mathcal{E}_c(\mathbf{R}) \end{aligned} \tag{34}$$

where ϕ satisfies Eqn. 32, the second term denotes the self energy of the nuclei and $\mathcal{E}_c(\mathbf{R})$ is the correction [33, 89] to the repulsive energy E_{zz} which is explained below. In above equations, the summation index J runs over all of the atoms in $\mathbb{R}^3$.

Energy correction Large-scale electronic structure calculations typically employ the pseudopotential approximation. Even though this is the case, the repulsive energy is calculated with the nuclei treated as point charges. This distinction is not made

by the formulation of electrostatics employed in this work, resulting in disagreement with conventional methods if the nuclear charge densities overlap. The correction to the repulsive energy which restores agreement can be expressed as [33, 89]

$$\begin{aligned}\mathcal{E}_c(\mathbf{R}) &= \frac{1}{2}\int_\Omega \left(\tilde{b}(\mathbf{x},\mathbf{R})+b(\mathbf{x},\mathbf{R})\right) V_c(\mathbf{x},\mathbf{R})\,\mathrm{d}\mathbf{x} + \frac{1}{2}\sum_J \int_\Omega b_J(\mathbf{x},\mathbf{R}_J) V^J_{\mathrm{loc}}(\mathbf{x},\mathbf{R}_J)\,\mathrm{d}\mathbf{x} \\ &\quad -\frac{1}{2}\sum_J \int_\Omega \tilde{b}_J(\mathbf{x},\mathbf{R}_J)\tilde{V}_J(\mathbf{x},\mathbf{R}_J)\,\mathrm{d}\mathbf{x}\,, \end{aligned} \tag{35}$$

where $V_c(\mathbf{x},\mathbf{R})$ is the solution of the Poisson equation

$$\frac{-1}{4\pi}\nabla^2 V_c(\mathbf{x},\mathbf{R}) = \tilde{b}(\mathbf{x},\mathbf{R}) - b(\mathbf{x},\mathbf{R}) \tag{36}$$

subject to periodic boundary conditions. The potential $V_c(\mathbf{x},\mathbf{R})$ so calculated is accurate to within a constant, which can be determined by evaluating $\sum_J (V^J_{\mathrm{loc}}(\mathbf{x},\mathbf{R}_J) - \tilde{V}_J(\mathbf{x},\mathbf{R}_J))$ at any point in space. Here, the 'reference' charge density $\tilde{b}(\mathbf{x},\mathbf{R})$ is the superposition of non-overlapping spherically symmetric and compactly supported 'reference' charge densities $\tilde{b}_J(\mathbf{x},\mathbf{R}_J) = -\frac{1}{4\pi}\nabla^2\tilde{V}_J(\mathbf{x},\mathbf{R}_J)$, i.e., $\tilde{b}(\mathbf{x},\mathbf{R}) = \sum_J \tilde{b}_J(\mathbf{x},\mathbf{R}_J)$. For the results presented in this work, we have chosen the 'reference' potential $\tilde{V}$ to be that employed previously in the context of all-electron electrostatics [78].

2.1.3 Energy and forces

Once we solve the self-consistent problem in Eqn. 28 for $\mathcal{D}^*$, the ground-state energy and atomic forces can be computed [89]. The electronic ground-state free energy can be written as

$$\begin{aligned}\mathcal{F}_0(\mathbf{R}) &= 2\mathrm{Tr}(\mathcal{D}^*\mathcal{H}^*) + E_{xc}(\rho^*) + \frac{1}{2}\int_\Omega (b(\mathbf{x},\mathbf{R})-\rho^*(\mathbf{x}))\phi^*(\mathbf{x},\mathbf{R})\,\mathrm{d}\mathbf{x} \\ &\quad -\int_\Omega V_{xc}(\rho^*(\mathbf{x}))\rho^*(\mathbf{x})\,\mathrm{d}\mathbf{x} - \frac{1}{2}\sum_J\int_\Omega b_J(\mathbf{x},\mathbf{R}_J)V^J_{\mathrm{loc}}(\mathbf{x},\mathbf{R}_J)\,\mathrm{d}\mathbf{x} \\ &\quad +\mathcal{E}_c(\mathbf{R}) + 2\sigma\mathrm{Tr}\left(\mathcal{D}^*\log\mathcal{D}^* + (\mathcal{I}-\mathcal{D}^*)\log(\mathcal{I}-\mathcal{D}^*)\right)\,, \end{aligned} \tag{37}$$

where $\mathcal{H}^*$ and ϕ^* are as in Eqns. 29 and 32 with $\rho = \rho^*$. The superscript '*' is used to denote quantities at the electronic ground state. Thereafter, the force on the I^{th}

nucleus may be obtained as

$$\begin{aligned}\mathbf{f}_I &= -\frac{\partial \mathcal{F}_0(\mathbf{R})}{\partial \mathbf{R}_I} \\ &= \sum_{I'} \int_\Omega \nabla b_{I'}(\mathbf{x},\mathbf{R}_{I'})(\phi^*(\mathbf{x},\mathbf{R}) - V_{\rm loc}^{I'}(\mathbf{x},\mathbf{R}_{I'}))\,\mathrm{d}\mathbf{x} + \mathbf{f}_I^c - 2\mathrm{Tr}\left[\mathcal{D}^* \frac{\partial \mathcal{V}_{\rm nl}}{\partial \mathbf{R}_I}\right], \quad (38)\end{aligned}$$

where the summation index I' runs over the I^{th} atom and its periodic images. In the above equation, $\mathbf{f}_I^c$ is the correction due to overlapping nuclear charge densities ($\mathbf{f}_I^c \equiv \mathbf{0}$ for no overlap) [33, 89], the expression for which is given by

$$\begin{aligned}\mathbf{f}_I^c &= \frac{1}{2}\sum_{I'} \int_\Omega \Big[\nabla \tilde{b}_{I'}(\mathbf{x},\mathbf{R}_{I'}) \left(V_c(\mathbf{x},\mathbf{R}) - \tilde{V}_{I'}(\mathbf{x},\mathbf{R}_{I'})\right) \\ &\quad + \nabla b_{I'}(\mathbf{x},\mathbf{R}_{I'}) \left(V_c(\mathbf{x},\mathbf{R}) + V_{\rm loc}^{I'}(\mathbf{x},\mathbf{R}_{I'})\right) + \nabla V_{c,I'}(\mathbf{x},\mathbf{R}_{I'}) \left(\tilde{b}(\mathbf{x},\mathbf{R}) + b(\mathbf{x},\mathbf{R})\right) \\ &\quad + b_{I'}(\mathbf{x},\mathbf{R}_{I'}) \nabla V_{\rm loc}^{I'}(\mathbf{x},\mathbf{R}_{I'}) - \tilde{b}_{I'}(\mathbf{x},\mathbf{R}_{I'}) \nabla \tilde{V}_{I'}(\mathbf{x},\mathbf{R}_{I'})\Big]\,\mathrm{d}\mathbf{x}\,, \quad (39)\end{aligned}$$

where the summation index I' runs over the I^{th} atom and its periodic images, and

$$\nabla V_{c,I'}(\mathbf{x},\mathbf{R}_{I'}) = \nabla \tilde{V}_{I'}(\mathbf{x},\mathbf{R}_{I'}) - \nabla V_{\rm loc}^{I'}(\mathbf{x},\mathbf{R}_{I'}). \quad (40)$$

It is worth noting that the evaluation of the above energy (Eqn. 35) and force corrections are $\mathcal{O}(N)$. The nonlocal pseudopotential component of the atomic force as presented in Eqn. 38 is

$$\mathbf{f}_I^{nl} = 2\mathrm{Tr}\left(\mathcal{D}^* \frac{\partial \mathcal{V}_{\rm nl}}{\partial \mathbf{R}_I}\right). \quad (41)$$

Expanding the density operator and nonlocal pseudopotential operator:

$$\begin{aligned}\mathcal{D}^* &= \sum_k g_k^* \left|\psi_k^*\right\rangle \left\langle\psi_k^*\right|, \quad (42)\\ \mathcal{V}_{\rm nl} &= \sum_I \sum_{lm} \gamma_l^I \left|\chi_{lm}^I\right\rangle \left\langle\chi_{lm}^I\right|, \quad (43)\end{aligned}$$

the nonlocal pseudopotential force in Eqn. 41 can be written as

$$\begin{aligned}\mathbf{f}_I^{nl} &= 2\mathrm{Tr}\left(\sum_{lm}\sum_k \gamma_l^I g_k^* \left[\left|\frac{\partial \chi_{lm}^I}{\partial \mathbf{R}_I}\right\rangle \left\langle\chi_{lm}^I|\psi_k^*\right\rangle \left\langle\psi_k^*\right| + \left|\chi_{lm}^I\right\rangle \left\langle\frac{\partial \chi_{lm}^I}{\partial \mathbf{R}_I}|\psi_k^*\right\rangle \left\langle\psi_k^*\right|\right]\right) \\ &= 4\left(\sum_{lm}\sum_k \gamma_l^I g_k^* \left\langle\chi_{lm}^I|\psi_k^*\right\rangle \left\langle\psi_k^*\right|\frac{\partial \chi_{lm}^I}{\partial \mathbf{R}_I}\right\rangle\Big), \quad (44)\end{aligned}$$

where ψ_k^* are the ground-state orbitals, and g_k^* are the ground-state occupations. Rather than employ the above expression, we utilize a change of variables technique [47] to rewrite it as

$$\mathbf{f}_I^{nl} = 4\left(\sum_{lm}\sum_{k}\gamma_l^I g_k^* \langle\chi_{lm}^I|\psi_k^*\rangle \langle\nabla\psi_k^*|\chi_{lm}^I\rangle\right) = 4\mathrm{Tr}\left(\mathcal{V}_{\mathrm{nl}}^I\mathcal{D}'\right) , \tag{45}$$

where

$$\mathcal{D}' = \sum_k g_k^* |\nabla\psi_k^*\rangle \langle\psi_k^*| . \tag{46}$$

We have found that the nonlocal pseudopotential force expression in Eqn. 45 results in significantly more accurate forces from a numerical standpoint compared to Eqn. 44. This is a consequence of the orbitals typically being smoother than the projectors.

2.2 Spectral Quadrature

Mature codes are now available implementing a number of the key linear scaling ideas [101, 35, 100, 112, 74, 70, 12]. However, despite steady and substantial advances, significant challenges remain. Accuracy and stability of $\mathcal{O}(N)$ approaches remain ongoing concerns due to the need for additional computational parameters, subtleties in determining sufficient numbers and/or centers of localized orbitals, and limitations of underlying basis sets, among others [13]. In real-space representations, the calculation of accurate atomic forces, as required for structural relaxations and molecular dynamics, has been a particular concern in $\mathcal{O}(N)$ as well as $\mathcal{O}(N^3)$ scaling methods [96, 11]. Perhaps most importantly, due to the assumption of a band gap in the electronic structure, the application of existing methods to metallic systems remains an open question [13]. Furthermore, due to the complex communications patterns and load balance issues which arise, particularly in localized orbital formulations, efficient large-scale parallelization poses a significant challenge.

The Spectral Quadrature (SQ) method has been recently proposed for the $\mathcal{O}(N)$ solution of the Kohn-Sham equations [103]. In SQ, the required electronic density, energy, and atomic forces are expressed as integrals over projected densities of states and

related quantities. With the choice of Gauss quadrature for integration, the method becomes equivalent to the classical recursion method [45, 46], while for Clenshaw-Curtis quadrature, the Fermi operator expansion (FOE) [38, 39] in Chebyshev polynomials is recovered. Since no assumption is made regarding the presence or absence of a band gap in the electronic structure, the SQ approach is general and applicable to metals and insulators alike. The computational cost of SQ is, however, inversely proportional to temperature, whereby it has a larger prefactor for metallic systems at lower temperature. Nevertheless, the amenability of SQ to large-scale parallel computation (Sections 4.6, 5.4) stands to mitigate this cost. In this work, we focus on the Clenshaw-Curtis variant of SQ since the atomic forces can be efficiently calculated compared to Gauss SQ. Moreover, the need for orthogonalization in Gauss SQ can limit performance when high orders of quadrature are required due to large spectral widths of the Hamiltonian.

In this section, we explain the Clenshaw-Curtis SQ method [89] for the case of infinite-cell calculation in order to convey the idea. In Chapter 4, the method will be applied to both Γ-point as well as infinite-cell calculations. The objective is to solve Eqn. 28 in an $\mathcal{O}(N)$ fashion. Based on the *nearsightedness principle* [90], the electronic interactions only within a distance of say R_{cut} are sufficient to compute the quantity of interest at a point in space. This enables development of $\mathcal{O}(N)$ electronic structure methods, where the density matrix for the system over Ω has exponential decay in real-space for both insulators and metals at finite temperature [36, 10]. So to compute any quantity of interest at a given point $\mathbf{x}$ in Ω, it would suffice to use operators defined on $H^1(\Omega_{\mathbf{x}})$ which is a space of functions in $L^2(\Omega_{\mathbf{x}})$ that have their first derivatives also in $L^2(\Omega_{\mathbf{x}})$, where $\Omega_{\mathbf{x}}$ is a domain around point $\mathbf{x} \in \Omega$ with its boundary at a distance of R_{cut} from the point $\mathbf{x}$ and is subjected to zero Dirichlet boundary conditions. In every iteration of SCF, the pointwise density operator $\mathcal{D}_{\mathbf{x}}$

for each $\mathbf{x} \in \Omega$ is given by:

$$\mathcal{D}_{\mathbf{x}} = g(\mathcal{H}_{\mathbf{x}}, \mu, \sigma) = \left(1 + \exp\left(\frac{\mathcal{H}_{\mathbf{x}} - \mu\mathcal{I}_{\mathbf{x}}}{\sigma}\right)\right)^{-1}, \tag{47}$$

where μ is calculated such that $2\int_\Omega \mathcal{D}(\mathbf{x},\mathbf{x})\,\mathrm{d}\mathbf{x} \approx 2\int_\Omega \mathcal{D}_{\mathbf{x}}(\mathbf{x},\mathbf{x})\,\mathrm{d}\mathbf{x} = N_\Omega$. Here, the subscript $([]_{\mathbf{x}})$ indicates that the operators are defined on $H^1(\Omega_{\mathbf{x}})$, where $\Omega_{\mathbf{x}}$ (as defined previously) is a localized domain around the point $\mathbf{x} \in \Omega$. The electron density is given by $\rho(\mathbf{x}) \approx 2\mathcal{D}_{\mathbf{x}}(\mathbf{x},\mathbf{x})$ for $\mathbf{x} \in \Omega$.

The pointwise density operator $(\mathcal{D}_{\mathbf{x}})$ is a function of the pointwise Hamiltonian operator $\mathcal{H}_{\mathbf{x}}$. Solving for electron density involves evaluating the diagonal components $(\mathbf{x},\mathbf{x})$ of the pointwise density operator for all $\mathbf{x} \in \Omega$. This can be computed by using Clenshaw-Curtis SQ method, where we expand any function of the pointwise Hamiltonian, $f(\mathcal{H}_{\mathbf{x}})$, in the Chebyshev polynomial basis. We shift and scale the pointwise Hamiltonian operator $\mathcal{H}_{\mathbf{x}}$ to $\hat{\mathcal{H}}_{\mathbf{x}} = (\mathcal{H}_{\mathbf{x}} - \chi_{\mathbf{x}}\mathcal{I}_{\mathbf{x}})/\zeta_{\mathbf{x}}$ such that the spectrum of $\hat{\mathcal{H}}_{\mathbf{x}}$ lies in $[-1,1]$. Here, $\chi_{\mathbf{x}} = (\lambda_{\mathbf{x}}^{max} + \lambda_{\mathbf{x}}^{min})/2$ and $\zeta_{\mathbf{x}} = (\lambda_{\mathbf{x}}^{max} - \lambda_{\mathbf{x}}^{min})/2$, with $\lambda_{\mathbf{x}}^{max}$ and $\lambda_{\mathbf{x}}^{min}$ denoting the maximum and minimum eigenvalues of $\mathcal{H}_{\mathbf{x}}$, respectively. Now, the Chebyshev expansion of order n_{pl} for any function $f : [-1,1] \to \mathbb{R}$ can be expressed as

$$f(\hat{\mathcal{H}}_{\mathbf{x}}) = \sum_{j=0}^{n_{pl}} f_j T_j(\hat{\mathcal{H}}_{\mathbf{x}})\,, \tag{48}$$

where T_j denote the Chebyshev polynomials of degree j and the expansion coefficients can be evaluated for given $f(r)$ as

$$f_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(r)T_j(r)}{\sqrt{1-r^2}}\,\mathrm{d}r\,, \quad j = 0,\ldots,n_{pl}\,, \tag{49}$$

with f_0 further scaled by a factor of half.

2.3 DFT outline

The solution to the fixed-point problem (Eqn. 28) in DFT is obtained through a self-consistent field (SCF) iteration. The outline of the SCF iteration which uses

the Spectral Quadrature method to compute electron density, is presented in Fig. 2. In this procedure, we start off with an initial guess for the electron density of the system in Ω, from which the electrostatic potential $\phi(\mathbf{x})$ can be computed by solving Poisson's equation (Eqn. 32). The effective potential $V_{\text{eff}} = V_{xc}(\rho) + \phi(\mathbf{x})$ can then be computed. This gives the updated information for the new Hamiltonian. Using this, we find the updated electron density from Clenshaw-Curtis Spectral Quadrature approach and the iteration repeats until convergence. Typically, the SCF convergence is accelerated using some kind of extrapolation (also called mixing) scheme (Chapter 3). Once the fixed-point iteration converges, energy and forces on the atoms can be evaluated.

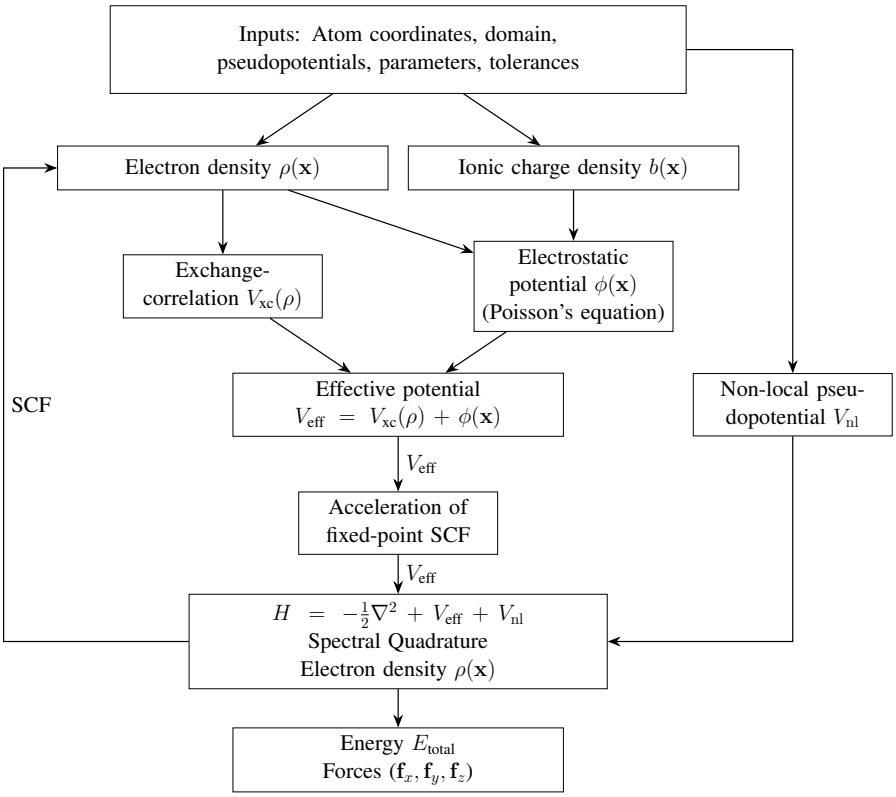


Figure 2: Outline of the DFT calculation to compute energy and forces using Spectral Quadrature method.

CHAPTER III

ACCELERATION OF FIXED POINT ITERATIONS

In this chapter, we discuss some acceleration techniques for fixed-point iterations. We propose two techniques — restarted Pulay and Alternating Anderson — which have shown efficient performance to accelerate linear as well as non-linear fixed point iterations. Specifically, we propose the restarted Pulay technique for SCF iteration and develop the restarted Pulay Jacobi linear solver by applying this technique to the Jacobi iteration used to solve linear system of equations. We propose the Alternating Anderson Jacobi linear solver by applying the Alternating Anderson extrapolation technique to the Jacobi iteration. We adopt the Periodic Pulay method proposed by [8] for accelerating the SCF. Periodic Pulay method is obtained by applying Alternating Anderson technique to the SCF iteration. We note that in the context of acceleration of SCF iteration, we use the words 'extrapolation', 'mixing' and 'acceleration' interchangeably. In this chapter, we first present some context for non-linear and linear fixed point iterations. Later, we discuss the extrapolation techniques and validate and compare their performance through examples.

3.1 Non-linear fixed point iterations

A non-linear fixed-point problem can be expressed as

$$\mathbf{x} = \mathbf{g}(\mathbf{x}) \,, \tag{50}$$

where $\mathbf{g} : \mathbb{C}^{N\times 1} \to \mathbb{C}^{N\times 1}$ represents the fixed-point mapping, with $\mathbb{C}$ denoting the set of all complex numbers. Fixed-point iterations are regularly encountered in a variety of scientific applications. Of particular interest in this work is the Self-Consistent Field (SCF) method [64], a standard approach for determining the electronic ground

state in ab-initio calculations like Density Functional Theory (DFT) [49, 54]. Since the computational time taken by electronic structure simulations is directly proportional to the number of SCF iterations required for convergence, there is great incentive in accelerating this process as far as possible [28]. Unfortunately, the rudimentary under-relaxed fixed-point iteration — commonly referred to as linear or simple mixing — typically converges extremely slowly, if at all. This is particularly true for large metallic systems at relatively low values of electronic temperature [63].

In view of the above discussion, a number of approaches have been proposed to accelerate the non-linear SCF fixed-point iteration. These include Pulay's Direct Inversion in the Iterative Subspace (DIIS) method [91] and its variants [14, 59], Broyden's quasi-Newton technique [18, 9] and its variations [102, 113, 27, 65], the Relaxed Constrained Algorithm (RCA) [19, 20], and a variety of preconditioning schemes [63, 51, 48, 93, 5]. Among these, Pulay's DIIS mixing scheme — based on the extrapolation method of Anderson [4] — has enjoyed considerable popularity and success due to its relative simplicity and overall performance [58]. Notably, the efficacy of Anderson's extrapolation scheme is not restricted to the SCF method alone [55], but also extends to a variety of other non-linear [30, 117, 33] and linear [4, 88] fixed-point problems. From a mathematical perspective, Pulay's technique can be considered to be a multisecant type method [28] that represents a specific variant of Broyden's approach [27].

The DIIS method is occasionally found to stagnate when employed in self-consistent electronic structure calculations, resulting in unacceptably slow or non-convergence. In an effort to overcome this, Fang and Saad [28] proposed performing a restart whenever the ratio between the current and previous iteration's residual exceeds a prespecified value. Additionally, some ab-initio codes provide the option of a periodic restart within Pulay mixing [6, 57]. However, these restart techniques introduce another parameter within the DIIS method, thereby adding further complexity to

the mixing scheme. In view of this, we are interested in a parameter-free restart strategy that not only prevents SCF iterations from stagnating, but also improves the efficiency and robustness of the DIIS method in general. To this end, we develop a variant of restarted Pulay for accelerating the convergence of fixed-point iterations. As an added bonus, the proposed approach is easily implementable within currently existing electronic structure codes.

3.2 Linear fixed point iterations

In nearly all areas of computational physics, it is common to encounter linear systems of equations of the form

$$\begin{aligned} &\mathbf{A}\mathbf{x} = \mathbf{b}\,, \\ &\mathbf{A} \in \mathbb{C}^{N\times N}\,,\ \ \mathbf{x} \in \mathbb{C}^{N\times 1}\ \text{ and }\ \mathbf{b} \in \mathbb{C}^{N\times 1}\,, \end{aligned} \tag{51}$$

where $\mathbb{C}$ is the set of all complex numbers. For small systems, solution strategies based on direct methods are typically the preferred choice. However, as the size of the system increases, it becomes necessary to employ iterative approaches in order to efficiently determine the solution. The basic fixed-point techniques that have been developed for this purpose include the Richardson, Jacobi, Gauss-Seidel, and Successive over-relaxation (SOR) methods [98]. However, these approaches suffer from relatively large prefactors and poor scaling with system size. This makes them unattractive for solving large systems of equations compared to Krylov subspace approaches such as the conjugate gradient [99] and Generalized Minimal Residual (GMRES) [97] methods.

In spite of the aforementioned limitations of basic fixed-point methods, the Jacobi iteration stands out because of its tremendous simplicity and potential for massive parallelization. This motivates the development of strategies that are able to significantly accelerate the convergence of the Jacobi method, while maintaining its underlying locality and simplicity to the maximum extent possible. Examples of

such approaches include the Chebyshev acceleration technique [98] and the recently proposed Scheduled Relaxation Jacobi (SRJ) method [118]. However, Chebyshev acceleration requires knowledge of the extremal eigenvalues of the matrix $\mathbf{A}$. Furthermore, the SRJ method as currently formulated is restricted to linear systems arising from second-order finite-difference discretization of elliptic equations. For such reasons, Krylov subspace techniques remain yet the methods of choice for the solution of large, sparse linear systems.

In this work, we explore the application of Anderson's (Pulay's DIIS) method to accelerate Jacobi linear fixed point iteration. In the linear setting, the DIIS approach bears remarkable similarity to the Generalized Minimal Residual (GMRES) method [97, 95, 114, 85].

3.3 Proposed methods

In the subsequent sections, we first discuss the original Anderson extrapolation (also called Pulay mixing) technique. We then, present the proposed restarted Pulay method. We demonstrate its efficiency for linear systems as well as SCF iterations in Section 3.5. Later, in Section 3.6, we present the Alternating Anderson Jacobi (AAJ) method proposed for linear systems. We study the performance of AAJ extensively as we choose to use it in the parallel $\mathcal{O}(N)$ DFT implementation due its massively parallelizable nature. For the same reason, we use the Periodic Pulay method to accelerate the SCF. Periodic Pulay method is not proposed as a part of this work and is discussed elsewhere [8].

3.4 Anderson extrapolation

Perhaps the simplest attempt at a solution to Eqn. 50 is an iteration of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}_k \,, \tag{52}$$

where $\mathbf{f}_k = (\mathbf{g}(\mathbf{x}_k) - \mathbf{x}_k)$ designates the residual, and $\beta \in \mathbb{C}$ signifies the relaxation parameter. In the context of electronic structure calculations, such an approach is referred to as linear or simple mixing. Depending on the spectral properties of the residual's Jacobian, the above iteration can converge extremely slowly, if at all [63]. The Anderson/Pulay method [4, 91] attempts to overcome this limitation by generalizing Eqn. 52 to

$$\mathbf{x}_{k+1} = \bar{\mathbf{x}}_k + \beta \bar{\mathbf{f}}_k \,, \tag{53}$$

where $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{f}}_k$ denote the normalized weighted averages of the previous $(m+1)$ iterates and residuals, respectively. Specifically,

$$\bar{\mathbf{x}}_k = \mathbf{x}_k - \sum_{j=1}^{m} \gamma_j \Delta\mathbf{x}_{k-m+j} \,, \tag{54}$$

$$\bar{\mathbf{f}}_k = \mathbf{f}_k - \sum_{j=1}^{m} \gamma_j \Delta\mathbf{f}_{k-m+j} \,, \tag{55}$$

where $\Delta\mathbf{x}_k = (\mathbf{x}_k - \mathbf{x}_{k-1})$, $\Delta\mathbf{f}_k = (\mathbf{f}_k - \mathbf{f}_{k-1})$, and the scalars $\Gamma_k = \begin{bmatrix} \gamma_1 & \gamma_2 & \dots & \gamma_m \end{bmatrix}^T \in \mathbb{C}^{m\times 1}$ are chosen so as to minimize the l_2-norm of the residual, i.e.,

$$\Gamma_k = \arg\min_{\Gamma_k} \|\bar{\mathbf{f}}_k\|^2 \,. \tag{56}$$

It can be shown that the optimized Γ_k satisfy the relation [28]

$$\left(\mathbf{F}_k^T \mathbf{F}_k\right) \Gamma_k = \mathbf{F}_k^T \mathbf{f}_k \,, \tag{57}$$

where the residual history

$$\mathbf{F}_k = \begin{bmatrix} \Delta\mathbf{f}_{k-m+1}, & \Delta\mathbf{f}_{k-m+2}, & \dots, & \Delta\mathbf{f}_k \end{bmatrix} \in \mathbb{C}^{N\times m} \,. \tag{58}$$

Thereafter, the update formula in Eqn. 53 takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta\mathbf{f}_k - (\mathbf{X}_k + \beta\mathbf{F}_k)(\mathbf{F}_k^T\mathbf{F}_k)^{-1}\mathbf{F}_k^T\mathbf{f}_k \,, \tag{59}$$

where the iterate history

$$\mathbf{X}_k = \begin{bmatrix} \Delta\mathbf{x}_{k-m+1}, & \Delta\mathbf{x}_{k-m+2}, & \dots, & \Delta\mathbf{x}_k \end{bmatrix} \in \mathbb{C}^{N\times m} \,. \tag{60}$$

In the above representations of $\mathbf{X}_k$ and $\mathbf{F}_k$, a zero or negative subscript indicates a null vector. Altogether, the parameters within Pulay's approach are the relaxation parameter β and the mixing history size $(m+1)$.

3.5 Restarted Pulay technique

The DIIS method described above utilizes the previous $(m+1)$ iterates for extrapolation after the starting $(m+1)$ iterations. Interestingly, while studying the performance of Anderson's extrapolation in the linear setting [88], we have discovered that introducing a specific type of periodic restart within the DIIS method significantly improves its performance [86]. In Algorithm 1, we outline the resulting restarted Pulay mixing variant, which we refer to as the r-Pulay method. In this technique, all but the last columns of $\mathbf{X}_k$ and $\mathbf{F}_k$ are cleared every $(m+1)$ iterations. This relatively subtle modification not only significantly improves the overall efficiency of Pulay's DIIS method, but also makes it noticeably more robust, as demonstrated by the examples that are discussed later. It is worth noting that since the restart frequency coincides with the mixing history size, r-Pulay does not have any parameters apart from those already existing in Pulay mixing.

Algorithm 1: Restarted Pulay (r-Pulay) method

Input: $\mathbf{x}_0$, β, m, tol, $\mathbf{X}_0 = [\,]$ and $\mathbf{F}_0 = [\,]$

repeat $k = 0, 1, 2 \ldots$

- $\mathbf{f}_k = \mathbf{g}(\mathbf{x}_k) - \mathbf{x}_k$
- **if** $k > 0$ **then**
  - **if** $k/(m+1) \in \mathbb{N}$ **then**
    - $\mathbf{X}_k = [\Delta\mathbf{x}_k], \quad \mathbf{F}_k = [\Delta\mathbf{f}_k]$
  - **else**
    - $\mathbf{X}_k = \begin{bmatrix} \mathbf{X}_{k-1}, & \Delta\mathbf{x}_k \end{bmatrix}, \quad \mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k-1}, & \Delta\mathbf{f}_k \end{bmatrix}$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \beta\mathbf{f}_k - (\mathbf{X}_k + \beta\mathbf{F}_k)(\mathbf{F}_k^T\mathbf{F}_k)^{-1}\mathbf{F}_k^T\mathbf{f}_k$
- **else**
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \beta\mathbf{f}_k$

until $\|f_k\| < tol$;

Output: $\mathbf{x}_k$

In addition to this work, there have been a few previous efforts directed at incorporating restarts within the Pulay mixing scheme. Specifically, Fang and Saad [28] proposed setting $\mathbf{X}_k = [\,]$ and $\mathbf{F}_k = [\,]$ whenever $\|\mathbf{f}_k\| < r_p\|\mathbf{f}_{k+1}\|$, r_p being the restart parameter. Additionally, some ab-initio codes like SIESTA [6] and PARSEC [57] provide the option of restarting the DIIS method at periodic intervals so as to overcome stagnating SCF iterations. In particular, the restart in SIESTA involves setting $\mathbf{X}_k = [\,]$ and $\mathbf{F}_k = [\,]$, and performing a linear mixing update in the subsequent iteration. However, unlike r-Pulay, the aforementioned restart strategies introduce an additional parameter into the DIIS method. Moreover, they do not retain the latest columns of $\mathbf{X}_k$ and $\mathbf{F}_k$, a feature found to have a significant impact on the performance.

3.5.1 Results and Discussion

We now verify the efficacy and accuracy of the proposed r-Pulay mixing scheme through selected examples. In sub-section 3.5.1.1, we test r-Pulay's ability to accelerate the classical Jacobi fixed-point iteration for the solution of large, sparse linear systems of equations arising in electronic structure simulations. Next, in sub-section 3.5.1.2, we study the effectiveness of r-Pulay in speeding-up the Self Consistent Field (SCF) method for Density Functional Theory (DFT) calculations. We perform all computations on a workstation with the following configuration: Intel Xeon Processor E3-1220 v3 (Quad Core, 3.10GHz Turbo, 8MB), 16GB (2x8GB) 1600MHz DDR3 ECC UDIMM.

3.5.1.1 Linear systems of equations: Accelerating the Jacobi iteration

Consider the following non-periodic Poisson and complex-valued periodic Helmholtz equations arising in real-space DFT [77, 106, 104, 78] and orbital-free DFT [71, 107, 33] simulations:

$$\textit{Ex1:} \qquad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) = \rho(\mathbf{r}) + b(\mathbf{r}) \quad \text{in } \Omega, \quad \Big\{ V(\mathbf{r}) = 0,\, \mathbf{r} \in \partial\Omega\,, \tag{61}$$

$$\textit{Ex2:} \quad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) + Q\,V(\mathbf{r}) = P\,\rho^{\alpha}(\mathbf{r}) \quad \text{in } \Omega, \quad \begin{cases} V(\mathbf{r}) = V(\mathbf{r} + L\hat{\mathbf{e}}_i),\, \mathbf{r} \in \partial\Omega\,, \\ \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r}) = \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r} + L\hat{\mathbf{e}}_i), \\ \mathbf{r} \in \partial\Omega\,, \end{cases} \tag{62}$$

where $\Omega \in \mathbb{R}^3$ is a cubical domain of side L with boundary $\partial\Omega$ and unit vectors $\hat{\mathbf{e}}_i$ aligned along the edges. The fields $\rho(\mathbf{r})$ and $b(\mathbf{r})$ denote the electron and nuclear charge densities respectively, obtained by the superposition of the corresponding isolated atom quantities [107, 105]. The constants $\alpha = \frac{5}{6} + \frac{\sqrt{5}}{6}$, $P = 0.0296 + i\,0.0217$ and $Q = -0.1284 - i\,0.1269$.

We discretize the aforementioned partial differential equations using sixth-order

accurate finite-differences. Specifically, we employ a mesh-size of $h = 0.5$ Bohr for the Poisson problem, with $\rho(\mathbf{r})$ and $b(\mathbf{r})$ corresponding to the Si_5H_{12}, $Si_{17}H_{36}$, $Si_{35}H_{36}$, $Si_{87}H_{76}$, $Si_{275}H_{172}$ and $Si_{525}H_{276}$ clusters. For the Helmholtz problem, we utilize $n_d = 45,\ 60,\ 75,\ 90,\ 120$, and 140 finite-difference nodes in each direction, with $\rho(\mathbf{r})$ corresponding to a system consisting of a vacancy in $3 \times 3 \times 3$ unit cells of Aluminum with lattice constant of 7.65 Bohr. The resulting linear systems of equations can be compactly written as

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,; \quad \mathbf{A} \in \mathbb{C}^{N\times N}\,, \mathbf{x} \in \mathbb{C}^{N\times 1} \ \text{and} \ \mathbf{b} \in \mathbb{C}^{N\times 1}\,, \tag{63}$$

where $\mathbf{A}$ is a sparse matrix that is symmetric positive-definite for the Poisson problem, and complex-symmetric for the Helmholtz problem. We solve these linear systems in the framework of the classical Jacobi iteration [98] (see sub-section 3.6.1), wherein the fixed-point mapping

$$\mathbf{g}(\mathbf{x}) = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x})\,, \tag{64}$$

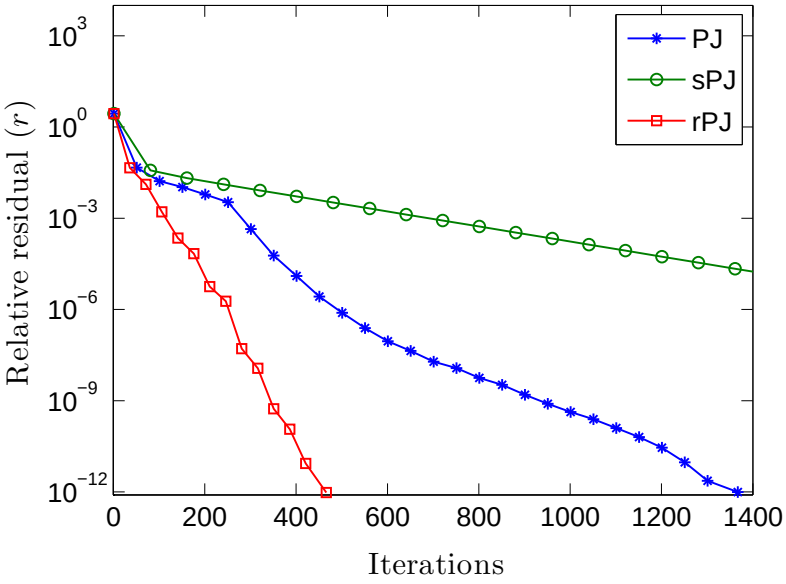
with $\mathbf{D}$ and $\mathbf{R}$ containing the diagonal and off-diagonal components of $\mathbf{A}$, respectively. We pick a vector of all ones as the starting guess $\mathbf{x}_0$, and set $tol = 1 \times 10^{-8}$ as the tolerance for convergence of the relative residual defined as

$$r_k = \frac{\|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|}{\|\mathbf{b}\|} = \frac{\|\mathbf{f}_k\|}{\|\mathbf{D}^{-1}\mathbf{b}\|}\,. \tag{65}$$

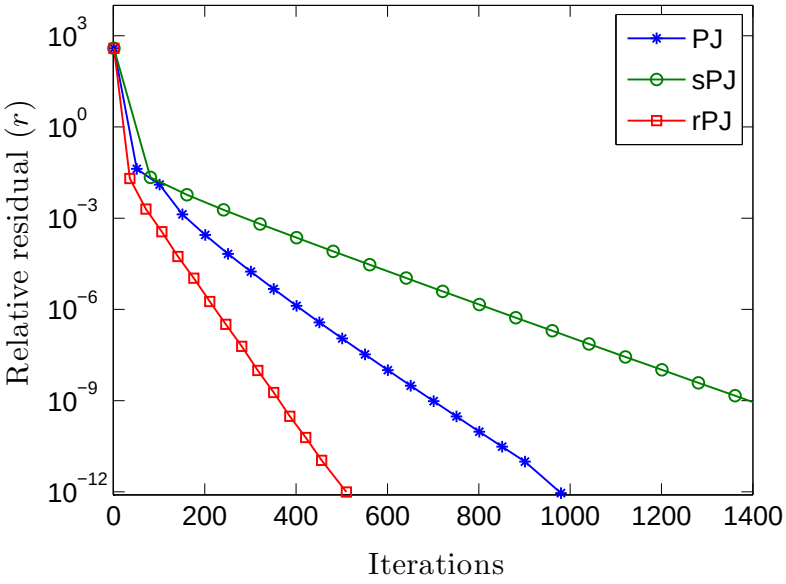
In the ensuing discussion, we shall refer to the Pulay accelerated Jacobi iteration as the Pulay-Jacobi (PJ) approach, the SIESTA restarted version with restarts performed at the $(k+1)/(m+2) \in \mathbb{N}$ iterations as sPJ, and the r-Pulay variant as the rPJ method.

First, we compare the reduction of the relative residual for the PJ, sPJ and rPJ methods in Fig. 3. We select *Ex1a* (Si_5H_{12} with $h = 0.5$ Bohr), and *Ex2e* (vacancy in $3 \times 3 \times 3$ unit cells of Aluminum with $n_d = 120$) as representative examples. We choose the parameters $\{\beta, m\} = \{0.5, 3\}$, which we have found to be close to optimal

for PJ in the context of the systems considered here. We observe that rPJ demonstrates significantly faster convergence than PJ, even though the chosen parameters are optimal for PJ and not rPJ. Additionally, rPJ is able to achieve extremely high accuracies while maintaining an elevated rate of convergence throughout the iteration. Indeed, rPJ's performance can be further enhanced with more judicious choice of parameters. We also note that sPJ demonstrates much slower convergence than the other two methods, which is a representative result for the linear systems considered here. In view of this, we will focus on the relative performance of PJ and rPJ for the remainder of this subsection.



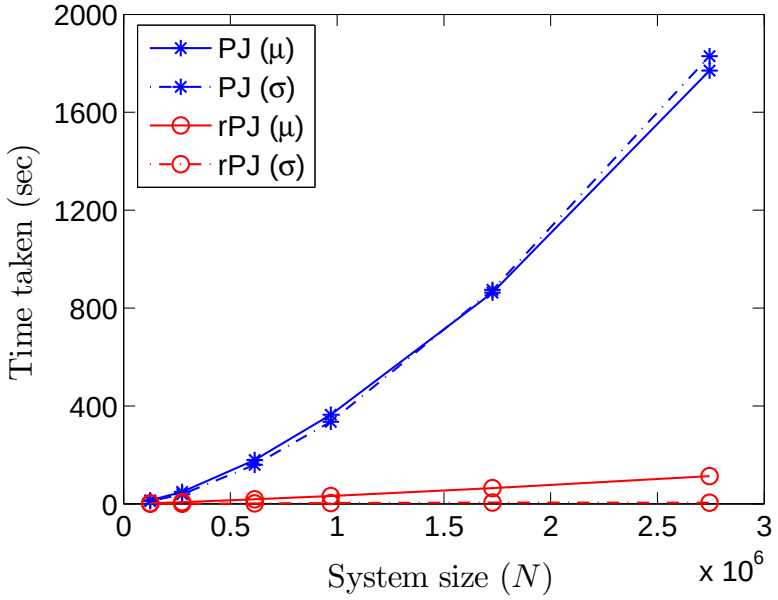
(a) *Ex1a*



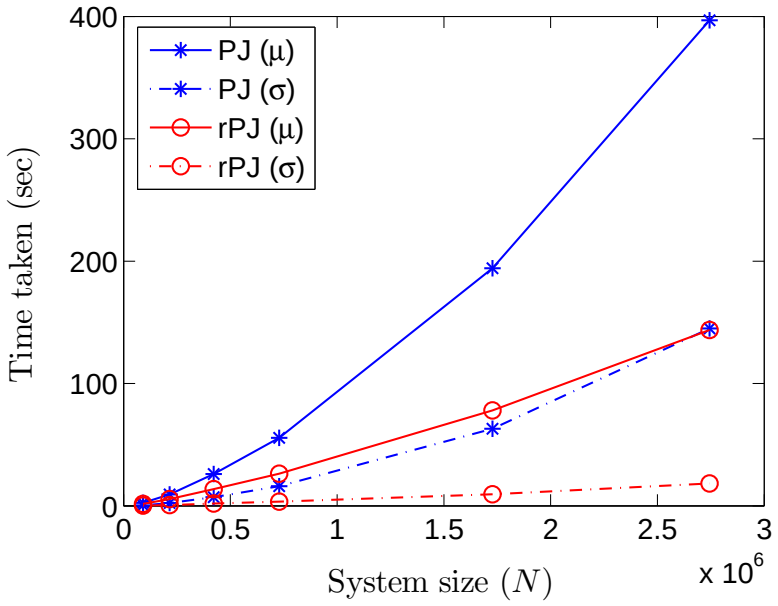
(b) *Ex2e*

Figure 3: Comparison of the convergence of the PJ, sPJ and rPJ methods. *Ex1a* signifies the Poisson equation for a Si_5H_{12} cluster with $h = 0.5$ Bohr. *Ex2e* denotes the Helmholtz equation for a vacancy in $3\times3\times3$ unit cells of Aluminum with $n_d = 120$.

Next, in Fig. 4, we compare the computational time taken by the PJ and rPJ methods for all the aforedescribed linear systems of equations. Specifically, we present the mean (μ) and standard deviation (σ) of the time taken for the parameters $\{\beta, m\} = \{0.5, 2 \text{ to } 8\}$. We observe that PJ's mean and standard deviation are noticeably larger than those of rPJ. In fact, for the biggest system in *Ex1*, PJ has larger μ and σ by factors exceeding 15 and 396, respectively. For the biggest system in *Ex2*, the corresponding ratios are close to 3 and 8, respectively. Remarkably, even though the Jacobi iteration is highly inefficient compared to Krylov subspace methods [98], rPJ is faster than the Generalized Minimal Residual Method (GMRES) [97] with a restart of 30 by factors exceeding 12 and 3 for the largest systems in *Ex1* and *Ex2*, respectively. This highlights the potential of rPJ as an efficient linear solver, particularly for large, sparse systems of equations. Overall, we conclude that rPJ represents an accelerated and significantly more robust version of PJ. Moreover, we expect that the proposed restart strategy will also be effective in the case of non-linear fixed-point problems, particularly as the iteration proceeds towards convergence.



(a) *Ex1*



(b) *Ex2*

Figure 4: Comparison of the performance of the PJ and rPJ methods. The mean and standard deviation are denoted by μ and σ, respectively. The linear systems of equations have been obtained from the discretization of the non-periodic Poisson and complex-valued periodic Helmholtz equations.

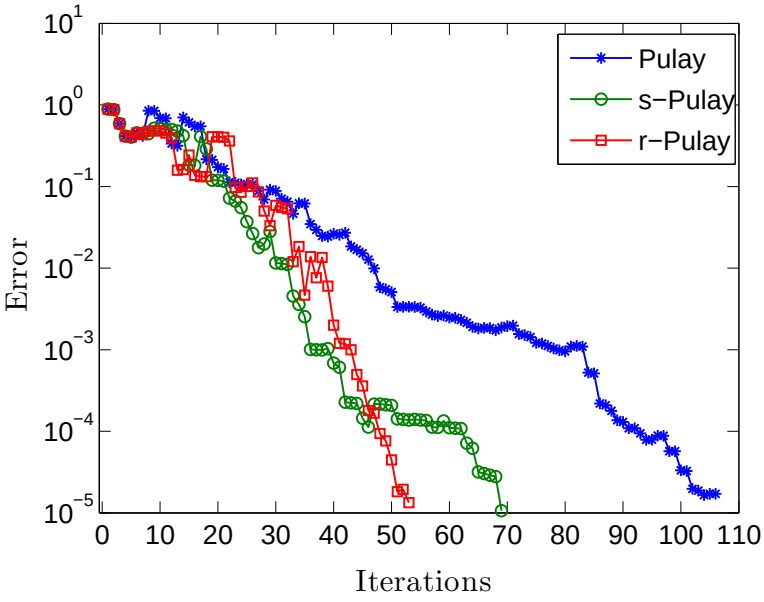
3.5.1.2 Density Functional Theory (DFT): Accelerating the Self Consistent Field (SCF) method

In this section, we study the efficacy of r-Pulay mixing in accelerating the convergence of the SCF method for DFT calculations. The SCF approach — one of the most commonly employed techniques for determining the electronic ground state in first principles calculations [55] — represents a non-linear fixed-point iteration with respect to either the electron density or the effective potential. The corresponding fixed-point mapping $\mathbf{g}(\mathbf{x})$ comprises of the electron density calculation given a Hamiltonian and effective potential evaluation given the electron density [64, 63]. Here, we perform all simulations in the framework provided by the quantum chemistry software SIESTA [101, 6]. Additionally, we denote the SIESTA variant of restarted Pulay with restarts performed at the $(k+1)/(m+2) \in \mathbb{N}$ iterations as the s-Pulay method.

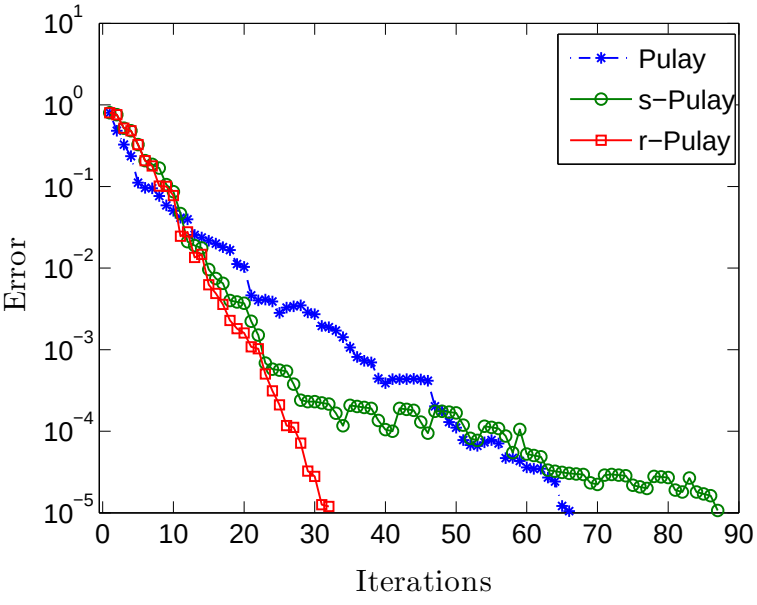
In order to ensure that the results presented here are easily reproducible, we consider examples that are available as test cases within the SIESTA distribution. Specifically, we focus on the following systems: (i) *sic-slab*: 78 atom silicon carbide surface saturated by Hydrogen. (ii) *ptcda*: 2 molecules of 3, 4, 9, 10 perylenetetracarboxylic dianhydride, consisting of 76 atoms. (iii) *fe_clust_noncollinear*: 3 atom iron cluster with noncollinear spin. (iv) *batio3*: 5 atom single unit cell of barium titanate. (v) *carbon_nanoscroll*: 140 atom carbon nanoscroll saturated with Hydrogen. (vi) *si001*: 10 atom (001) Silicon surface saturated with Hydrogen. (vii) *si111-spinpol*: 22 atom (111) Silicon surface saturated with Hydrogen. In all of these examples, the only modifications made to the input files are enabling of spin polarized calculations, varying the Pulay mixing history, and using SIESTA's default tolerances for convergence of the SCF method. The motivation for including spin is that typically larger number of iterations are required for achieving convergence, which makes acceleration of the SCF process even more desirable.

We start by comparing the convergence of the Pulay, s-Pulay and r-Pulay methods

during the SCF iteration. Selecting *sic-slab* and *ptcda* as representative examples, we plot the error as a function of iteration number in Fig. 5. Here, error denotes the maximum difference (in magnitude) between the density matrix of two consecutive SCF iterations. We have employed mixing history of 5 ($m = 4$) for both *sic-slab* and *ptcda*. We observe that both r-Pulay and s-Pulay converge faster than Pulay for the *sic-slab* system. In fact, r-Pulay and s-Pulay demonstrate similar performance up to an error of around 10^{-4}, after which s-Pulay experiences a noticeable reduction in the convergence rate. The trends are similar for *ptcda*, with s-Pulay's drop in convergence rate so dramatic that it requires larger number of iterations than even Pulay to reduce the error to 10^{-5}. Altogether, r-Pulay is found to be the most efficient, and is able to achieve practical SCF tolerances in nearly half the iterations needed by Pulay mixing. This is consistent with previous results obtained in the linear setting for the Jacobi fixed-point iteration.



(a) sic-slab



(b) ptcda

Figure 5: Progression of the error during the SCF iteration for the Pulay, s-Pulay and r-Pulay methods. Error denotes the maximum difference (in magnitude) between the density matrix of two consecutive SCF iterations.

Next, we compare the performance of the Pulay, s-Pulay and r-Pulay methods for the seven aforementioned electronic structure problems. Specifically, we determine the number of SCF iterations required to achieve the SIESTA default tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy for $m = 2, 3$ and 4. Recall that $(m + 1)$ denotes the number of iterates used for extrapolation. The results so obtained are presented in Table 1. We observe that both r-Pulay and s-Pulay are significantly more efficient and robust versions of the Pulay method, with r-Pulay demonstrating the best performance overall. In particular, r-Pulay is relatively insensitive to the amount of mixing history, whereas large variations can be seen in the performance of the DIIS method. Furthermore, the proposed restart is able to speed-up Pulay mixing by up to factors exceeding 3. In fact, even for the optimal choice of $2 \leq m \leq 4$ within the Pulay method, r-Pulay demonstrates superior performance by up to factors nearing two. Intriguingly, s-Pulay consistently demonstrates superior performance to Pulay even though the opposite trend was observed in the linear setting.

Finally, we study the statistics of the number of SCF iterations required for convergence when $2 \leq m \leq 8$. In Table 2, we present the mean (μ) and standard deviation (σ) for the Pulay, s-Pulay and r-Pulay methods. We observe that r-Pulay demonstrates the best performance among the three approaches. Specifically, r-Pulay possesses the smallest values of mean and standard deviation, further highlighting its efficiency and robustness. It is worth emphasizing that even though we have focused on spin polarized calculations, the above inferences are applicable to systems where spin is neglected. Consider for example, the systems *sic-slab* and *ptcda*. In the case of *sic-slab*, $\{\mu, \sigma\} = \{100, 43\}$, $\{60, 14\}$ and $\{47, 3\}$ for the Pulay, s-Pulay and r-Pulay methods, respectively. For *ptcda*, the corresponding numbers are $\{50, 28\}$, $\{30, 8\}$ and $\{26, 2\}$, respectively. It is clear that our previous conclusions are still valid.

As part of this work, we have performed a variety of simulations — including

Table 1: Number of SCF iterations taken by the Pulay, s-Pulay and r-Pulay methods to achieve the default SIESTA convergence tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy. Number of iterates that have been utilized for mixing is $(m+1)$.

System	$m=2$		
	Pulay	s-Pulay	r-Pulay
sic-slab	104	59	52
ptcda	93	40	30
fe_clust_noncollinear	145	79	88
batio3	146	47	39
carbon_nanoscroll	39	22	21
si001	29	24	21
si111-spinpol	41	27	21
	$m=3$		
sic-slab	119	59	46
ptcda	53	34	38
fe_clust_noncollinear	93	98	61
batio3	137	38	38
carbon_nanoscroll	34	20	21
si001	26	24	22
si111-spinpol	27	24	24
	$m=4$		
sic-slab	101	63	59
ptcda	59	46	28
fe_clust_noncollinear	187	160	71
batio3	61	35	34
carbon_nanoscroll	35	23	23
si001	21	22	19
si111-spinpol	21	24	20

a number of systems not presented here — to establish the relative performance of Pulay, s-Pulay and r-Pulay. In all of these examples, we have found r-Pulay to be significantly more efficient and robust compared to Pulay. Occasionally, we have noticed the performance of s-Pulay to be slightly better than r-Pulay. As an example, for the *nanotube-c-5-0* (20 atom C(5,0) nanotube) system, $\{\mu,\sigma\}=\{21,2\}$ and $\{24,6\}$ for the s-Pulay and r-Pulay methods, respectively. Such results have been observed when Pulay itself requires relatively few iterations for convergence. However, for systems

Table 2: Statistics of the number of SCF iterations required for convergence when m takes values in the range of 2 to 8. The mean and standard deviation are denoted by μ and σ, respectively. The default SIESTA convergence tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy have been employed.

System	Pulay		s-Pulay		r-Pulay	
	μ	σ	μ	σ	μ	σ
sic-slab	87	21	69	13	56	6
ptcda	49	22	46	9	32	4
fe_clust_noncollinear	198	108	124	34	78	25
batio3	75	48	42	11	37	3
carbon_nanoscroll	30	6	22	2	21	1
si001	23	4	22	2	20	1
si111-spinpol	25	8	23	3	21	2

where convergence of the SCF is challenging, we have found that r-Pulay outperforms s-Pulay. Overall, we conclude that r-Pulay is a viable and attractive method for accelerating the SCF iteration in electronic structure calculations.

3.6 Alternating Anderson technique

In this work, we treat the Jacobi method as a fixed-point iteration and employ Anderson's extrapolation to accelerate its convergence. However, rather than applying the extrapolation in every step, we employ it at periodic intervals within the Jacobi iteration. We refer to this approach as the Alternating Anderson Jacobi (AAJ) method. We verify the accuracy, efficiency, and generality of AAJ in a range of test cases, including nonsymmetric, 3D Poisson, and complex-valued Helmholtz problems. In particular, we demonstrate that AAJ is able to accelerate the classical Jacobi method by factors exceeding $10,000$, and substantially outperform GMRES in the process.

3.6.1 Jacobi method

Consider the linear system of equations described by Eqn. 63. The matrix $\mathbf{A}$ can be split as

$$\mathbf{A} = \mathbf{D} + \mathbf{R}\,, \tag{66}$$

where all the off-diagonal components of the matrix $\mathbf{D} \in \mathbb{C}^{N\times N}$ and the diagonal components of $\mathbf{R} \in \mathbb{C}^{N\times N}$ are identically zero. Using this decomposition, Eqn. 63 can be rewritten as the fixed-point problem

$$\mathbf{x} = \mathbf{g}(\mathbf{x})\,, \tag{67}$$

where the mapping

$$\mathbf{g}(\mathbf{x}) = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x})\,. \tag{68}$$

In this setting, the residual can be defined to be

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \mathbf{x}\,. \tag{69}$$

Further, the error

$$\mathbf{e}(\mathbf{x}) = \mathbf{x} - \mathbf{x}^*\,, \tag{70}$$

where $\mathbf{x}^*$ denotes the solution of the linear system in Eqn. 63. This Jacobi-type reformulation is predicated on the assumption that there are no zeros on the diagonal of $\mathbf{D}$, and therefore by extension the diagonal of $\mathbf{A}$.

The Jacobi method [98, 92] proposes to solve the fixed-point problem in Eqn. 67 using an iteration of the form

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k)\,, \tag{71}$$

where the subscript k is used to represent the iteration number. In this approach, the relationship between the error/residual in any two consecutive iterates can be shown to be

$$\mathbf{e}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\right)\mathbf{e}(\mathbf{x}_k)\,, \quad \mathbf{f}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\right)\mathbf{f}(\mathbf{x}_k)\,. \tag{72}$$

It follows that the Jacobi method is effective at nullifying the error/residual components corresponding to eigenvalues of $(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A})$ whose magnitudes are close to zero, and relatively ineffective at nullifying components corresponding to eigenvalues with magnitudes near but less than unity. In particular, convergence of the Jacobi

iteration requires

$$\|\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\| < 1\,, \tag{73}$$

where $\|.\|$ refers to the 2-norm. Such a constraint limits the applicability of the Jacobi method, which motivates suitable modification of the underlying iteration.

The Weighted Jacobi (WJ) method [98, 92] represents a generalization of the aforedescribed Jacobi technique, wherein the fixed-point iteration in Eqn. 71 takes the form

$$\mathbf{x}_{k+1} = (1-\omega)\mathbf{x}_k + \omega\mathbf{g}(\mathbf{x}_k)\,. \tag{74}$$

In terms of the residual, the above equation reduces to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \omega\mathbf{f}(\mathbf{x}_k)\,. \tag{75}$$

The scalar $\omega \in \mathbb{C}$ is referred to as the relaxation parameter, with the specific choice of $\omega = 1$ yielding the standard Jacobi iteration. Analogous to the Jacobi method, the progression of error/residual in the WJ iteration can be expressed as

$$\mathbf{e}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A}\right)\mathbf{e}(\mathbf{x}_k)\,, \quad \mathbf{f}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A}\right)\mathbf{f}(\mathbf{x}_k)\,. \tag{76}$$

It follows that the WJ approach is efficient for error/residual components corresponding to eigenvalues of $(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A})$ whose magnitudes are near zero, while relatively inefficient at reducing components corresponding to eigenvalues with magnitudes close to but less than one. Furthermore, convergence of the WJ method requires

$$\|\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A}\| < 1\,. \tag{77}$$

Overall, when $\mathbf{D}^{-1}\mathbf{A}$ has eigenvalues with positive real part, an appropriately small relaxation parameter ω can be chosen to enable convergence when the standard Jacobi method diverges. However, doing so negatively impacts the performance of the WJ method in neutralizing error/residual components corresponding to small-magnitude eigenvalues of $\mathbf{D}^{-1}\mathbf{A}$. A detailed description and analysis of the classical Jacobi method and its weighted counterpart can be found in standard texts [98, 92, 41, 115].

3.6.2 Anderson-Jacobi method

Anderson demonstrated in his original work [4] that the proposed extrapolation technique can be employed to significantly improve the performance of the Jacobi method, among others. In spite of this, such an approach for solving linear systems of equations—which we shall refer to as the Anderson-Jacobi (AJ) method—has received little attention subsequently. In this work, we demonstrate that AJ is an efficient method for solving large systems of equations. In fact, as shown in subsection 3.6.4, it is able to consistently outperform GMRES for all the cases considered here.

In the AJ method, the fixed-point iteration in Eqn. 75 is generalized to

$$\mathbf{x}_{k+1} = \bar{\mathbf{x}}_k + \beta \mathbf{f}(\bar{\mathbf{x}}_k)\,, \tag{78}$$

where $\bar{\mathbf{x}}_k$ denotes the weighted average of the previous iterates and $\beta \in \mathbb{C}$ is a parameter. Specifically,

$$\bar{\mathbf{x}}_k = \mathbf{x}_k - \sum_{j=1}^{m} \gamma_j \left(\mathbf{x}_{k-m+j} - \mathbf{x}_{k-m+j-1}\right)\,, \tag{79}$$

where $m+1$ is the number of iterates used for extrapolation. The update formula then becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}(\mathbf{x}_k) - (\mathbf{X}_k + \beta \mathbf{F}_k)(\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T \mathbf{f}(\mathbf{x}_k)\,. \tag{80}$$

The aforedescribed AJ approach can also be interpreted as a multi-secant type method [28, 65]. In this context, the AJ iteration generalizes Eqn. 75 to take the form

$$\mathbf{x}_{k+1} \;=\; \mathbf{x}_k + \mathbf{C}_k \mathbf{f}(\mathbf{x}_k)\,. \tag{81}$$

The matrix $\mathbf{C}_k \in \mathbb{C}^{N \times N}$ is set to the solution of the constrained minimization [28, 63]

$$\begin{aligned} &\min_{\mathbf{C}_k} \|\mathbf{C}_k + \beta \mathbf{I}\| \\ \text{s.t.} \quad &\mathbf{C}_k \mathbf{F}_k = \mathbf{X}_k\,, \end{aligned} \tag{82}$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix, with $\mathbb{R}$ denoting the set of all real numbers. The solution to this variational problem is

$$\mathbf{C}_k \;=\; \beta\mathbf{I} - (\mathbf{X}_k + \beta\mathbf{F}_k)(\mathbf{F}_k^T\mathbf{F}_k)^{-1}\mathbf{F}_k^T\,. \tag{83}$$

On substituting this expression for $\mathbf{C}_k$ into Eqn. 81, the AJ fixed-point iteration in Eqn. 80 is recovered. It is worth noting that the constraint in Eqn. 82 can be expressed as

$$(\mathbf{C}_k\mathbf{D}^{-1}\mathbf{A})\mathbf{X}_k = \mathbf{X}_k\,, \tag{84}$$

from which it can be inferred that $\mathbf{C}_k$ is designed to approximate $\mathbf{A}^{-1}\mathbf{D}$ in Anderson's extrapolation.

In the AJ method, the relation between the error/residual at consecutive iterations can be written as

$$\mathbf{e}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{C}_k\mathbf{D}^{-1}\mathbf{A}\right)\mathbf{e}(\mathbf{x}_k)\,, \quad \mathbf{f}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{C}_k\mathbf{D}^{-1}\mathbf{A}\right)\mathbf{f}(\mathbf{x}_k)\,. \tag{85}$$

From these equations, it can be deduced that the AJ update is a contraction provided

$$\|\mathbf{I} - \mathbf{C}_k\mathbf{D}^{-1}\mathbf{A}\| < 1\,. \tag{86}$$

Furthermore, the fixed-point iteration will converge faster when $\mathbf{C}_k$ is able to better approximate $\mathbf{A}^{-1}\mathbf{D}$. As a result, the AJ method significantly accelerates the convergence of the basic Jacobi iteration, as verified in sub-section 3.6.4. Overall, the AJ method can be viewed as a generalization of the Weighted Jacobi (WJ) method, since it replaces the constant matrix $\omega\mathbf{I}$ with a dynamically updated matrix $\mathbf{C}_k$. Furthermore, AJ reduces to WJ on setting $\beta = \omega$ and $m = 0$.

3.6.3 Alternating Anderson-Jacobi method

The weighted Jacobi method typically suffers from slow convergence due to its inability to efficiently reduce the 'low frequency' components of the error/residual. Here

and henceforth, 'low frequency' and 'high frequency' error/residual components denote those corresponding to the eigenvalues of $(\mathbf{I} - \omega \mathbf{D}^{-1}\mathbf{A})$ with magnitude close to unity and zero, respectively. In this work, we aim to develop an accelerated variant of the Jacobi method, while seeking to retain its tremendous simplicity, locality, and potential for scalability on massively parallel architectures. We shall refer to this generalization, which incorporates both Weighted Jacobi (WJ) and Anderson-Jacobi (AJ) updates, as the Alternating Anderson-Jacobi (AAJ) method [88].

In Fig. 6, we outline the algorithm of the proposed AAJ method. We have used $\mathbf{x}_0$ to represent the initial guess, r to denote the normalized l_2 norm of the residual, and ϵ to signify the tolerance specified for convergence. In the AAJ approach, the fixed-point iteration takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}_k \mathbf{f}(\mathbf{x}_k)\,, \tag{87}$$

where the matrix

$$\mathbf{B}_k = \begin{cases} \omega \mathbf{I} & \text{if } (k+1)/p \notin \mathbb{N}\,, \\ \beta \mathbf{I} - (\mathbf{X}_k + \beta \mathbf{F}_k)(\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T & \text{if } (k+1)/p \in \mathbb{N}\,. \end{cases} \tag{88}$$

In this setting, the relationship between the error/residual at consecutive iterations can be written as

$$\mathbf{e}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{B}_k \mathbf{D}^{-1}\mathbf{A}\right) \mathbf{e}(\mathbf{x}_k)\,, \quad \mathbf{f}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{B}_k \mathbf{D}^{-1}\mathbf{A}\right) \mathbf{f}(\mathbf{x}_k)\,. \tag{89}$$

It follows that the AAJ update is a contraction provided

$$\|\mathbf{I} - \mathbf{B}_k \mathbf{D}^{-1}\mathbf{A}\| < 1\,. \tag{90}$$

Overall, AAJ represents a generalization of the WJ method wherein the WJ update in Eqn. 75 is replaced with an AJ update described by Eqn. 80 every p^{th} iteration. It can also be viewed as a generalization of the AJ method in which $\mathbf{C}_k = \omega \mathbf{I}$ if $(k+1)/p \notin \mathbb{N}$. Indeed, the AJ method is recovered for $p = 1$ and the WJ method is recovered in the limit $p \to \infty$.

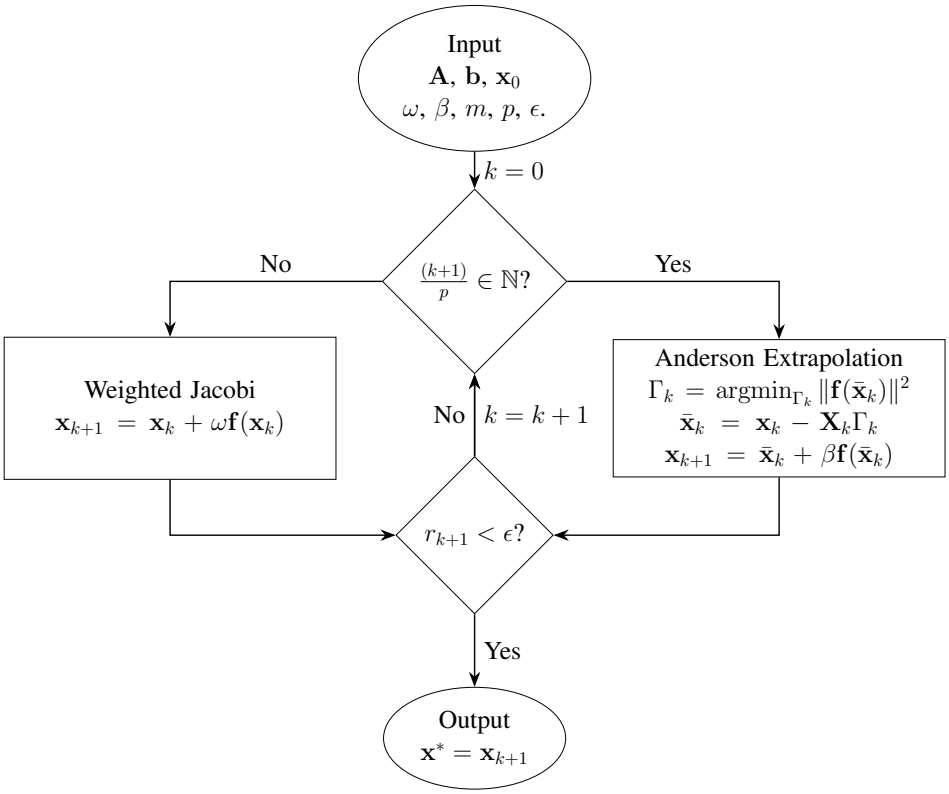


Figure 6: The Alternating Anderson-Jacobi (AAJ) method.

In this work, we have employed Anderson's extrapolation to accelerate the convergence of the Jacobi method. Indeed, we expect such an approach to be effective in the context of other stationary iterative methods, e.g., Richardson iteration, Gauss-Seidel, and Successive Over Relaxation (SOR). Notably, when the proposed approach is developed in the context of the Richardson iteration, the resulting technique—which we shall refer to as the Alternating Anderson-Richardson (AAR) method—represents a generalization of the approach proposed by Khabaza [52]. In particular, the AAR method will reduce to Khabaza's approach on setting $\omega = 1$, $\beta = 0$, and $p = m + 1$, with the coefficients Γ_k calculated only when the reduction in the residual is smaller than a specified threshold. Most significantly for parallel computing, the AAJ method is more amenable to efficient massively parallel implementation [108] than AJ, due to the reduction in global communication associated with the evaluation of Γ_k in every

AJ update.

3.6.4 Results and discussion

In this section, we validate the accuracy and effectiveness of the proposed Alternating Anderson-Jacobi (AAJ) approach in a series of test cases, including nonsymmetric, Poisson, and complex-valued Helmholtz problems. Using finite differences, we discretize the partial differential equations in a domain Ω having boundary $\partial\Omega$ with outward unit normal $\mathbf{n}$. We denote the mesh-size by h and the number of nodes in each direction by n_d. For the resulting linear systems of equations, we employ the following nomenclature for presenting results and ensuing discussion. We denote each partial differential equation and its associated boundary conditions by 'Problem #', where '#' represents a number. Next, we associate with every linear system a four character label, in which we abbreviate 'Problem #' in the first two characters as 'P#'. We set the third character as either 'a' or 'b', where 'a' denotes a collection of 'P#' systems having the same Ω with varying h, whereas 'b' denotes same h with varying Ω. We append a number as the fourth character to indicate the value of n_d, with values in ascending order. For example, if $n_d = n_1$, n_2, and n_3 ($n_1 < n_2 < n_3$) are used for discretization, fourth characters of '1', '2', and '3' signify systems with $n_d = n_1$, n_2, and n_3, respectively.

We compare the performance of the AAJ method with the Weighted Jacobi (WJ), Anderson-Jacobi (AJ), and Scheduled Relaxation Jacobi (SRJ) [118] fixed-point approaches. In the AJ and AAJ iterations, we employ the Moore-Penrose pseudoinverse [25] for the calculation of $(\mathbf{F}_k^T\mathbf{F}_k)^{-1}$ since $\mathbf{F}_k^T\mathbf{F}_k$ can become ill-conditioned as the iteration proceeds, for large m in particular. We also compare with the Krylov subspace method GMRES [97], whose efficiency can be significantly enhanced with sophisticated preconditioning schemes such as multigrid [44]. However, such schemes increase the cost per iteration, and pose significant challenges for large-scale parallelization

[118]. The aim of the present work is to retain as far as possible the simplicity and computational locality of the classical Jacobi method while substantially accelerating it without need of such advanced preconditioning, thus providing a method well-suited to large-scale parallel implementation. Hence, for the present purposes, we shall compare to GMRES with simple Jacobi preconditioning, using the same inverse diagonal as in the Jacobi iteration. It is worth noting however that, given the relation of Anderson and GMRES iterations [95], if more sophisticated preconditioners are available, they may be expected to benefit AJ and AAJ iterations as well as GMRES.

We perform all calculations using MATLAB [43] on a workstation with the following configuration: Intel Xeon Processor E3-1220 v3 (Quad Core, 3.10GHz Turbo, 8MB), 16GB (2x8GB) 1600MHz DDR3 ECC UDIMM.

3.6.4.1 Model problem: Laplace equation

The Laplace equation is among the most well studied partial differential equations, making it an excellent test case. We generate the corresponding linear systems using second-order finite-differences. Since the matrix $\mathbf{D}^{-1}\mathbf{A}$ is independent of the mesh-size h, we only consider systems resulting from varying h with fixed domain Ω. In the Weighted Jacobi (WJ) method, we utilize the optimal relaxation parameter [98] of $\omega = \omega^* = 2/(\lambda_1 + \lambda_N) = 1$, where λ_1 and λ_N are the minimum and maximum eigenvalues of $\mathbf{D}^{-1}\mathbf{A}$. In situations where ω^* results in a non-convergent iteration due to finite precision, we reduce ω^* by 0.01. In the Anderson-Jacobi (AJ) approach, we choose $\{\beta, m\} = \{0.2, 10\}$, which we have found to be efficient after an initial traversal of the two-dimensional parameter space. Since $\mathbf{b} = \mathbf{0}$ for the Laplace equation, it is not possible to use the relative residual for the stopping criterion. Instead, we define the normalized residual in this case as

$$r_{k+1} = \frac{\|\mathbf{f}(\mathbf{x}_k)\|}{\|\mathbf{f}(\mathbf{x}_0)\|}\,, \tag{91}$$

and set $\epsilon = 1 \times 10^{-8}$ as the tolerance for convergence. We employ the same random starting guess while studying the relative performance of different approaches.

One-dimensional Laplace equation We first consider the one-dimensional Laplace equation with zero Dirichlet and Neumann boundary conditions:

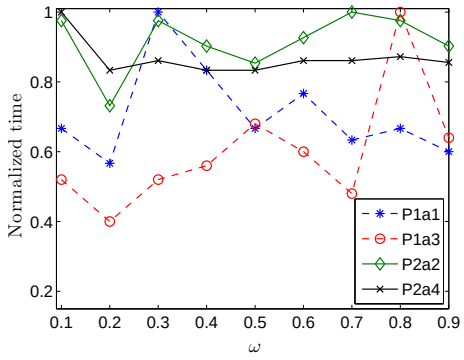
$$\text{Problem 1:} \qquad -V_{xx}(x) = 0 \text{ in } \Omega\,, \quad V(x) = 0 \text{ on } \partial\Omega\,, \tag{92}$$

$$\text{Problem 2:} \qquad -V_{xx}(x) = 0 \text{ in } \Omega\,, \quad V_x(x) = 0 \text{ on } \partial\Omega\,, \tag{93}$$

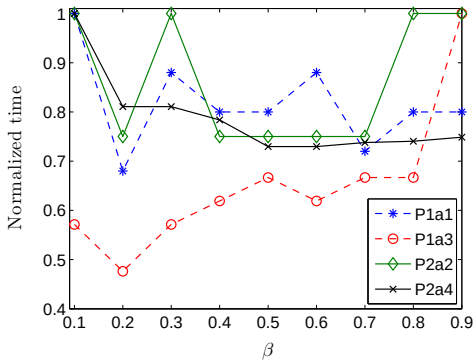
where $\Omega = (0, L)$. We choose a domain of size $L = 100$, and discretize it using $n_d = 101, 301, 1001, 3001$ and $10,001$ finite-difference nodes. The matrices $\mathbf{A}$ resulting from the discretization of Problems 1 and 2 are positive-definite and positive-semidefinite respectively. Further, their respective solutions are $\mathbf{x}^* = 0$ and $\mathbf{x}^* = \mathbf{c}$, where $\mathbf{c}$ is any constant vector.

The parameters within the AAJ method are $\{\omega, \beta, m, p\}$, where ω and β are the relaxation parameters in the WJ and AJ updates respectively, $m + 1$ is the number of iterates in the Anderson extrapolation history, and p is the frequency of the AJ update. We choose 'P1a1', 'P1a3', 'P2a2', and 'P2a4' as representative systems to perform a parametric study. After a preliminary traversal of the four-dimensional space of parameters, we have found $\{\omega, \beta, m, p\} = \{0.2, 0.2, 10, 6\}$ to be an efficient set. In Fig. 7, we present the normalized computational time taken when three of these parameters are fixed and the fourth one is varied. We observe that the performance of AAJ is relatively insensitive to the choice of ω and β. We find $m \sim 10$ to be optimal, with a steep increase in time for smaller values. We also notice a drastic reduction in performance for $p > 10$, with $p \sim 6$ being optimal. We have made similar observations for the Poisson and Helmholtz equations, with AAJ again relatively insensitive to the choice of parameters. Overall, we find $\{\omega, \beta, m, p\} = \{0.2, 0.2, 10, 6\}$ to perform appreciably, with solution times of 0.02, 0.10, 0.03, and 0.30 seconds for 'P1a1', 'P1a3', 'P2a2', and 'P2a4' systems, respectively. We shall employ this set of

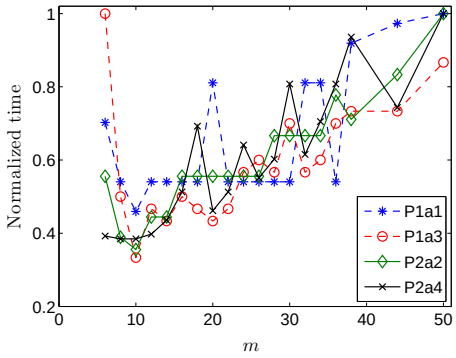
parameters within AAJ for the Laplace, Poisson, and Helmholtz equations.



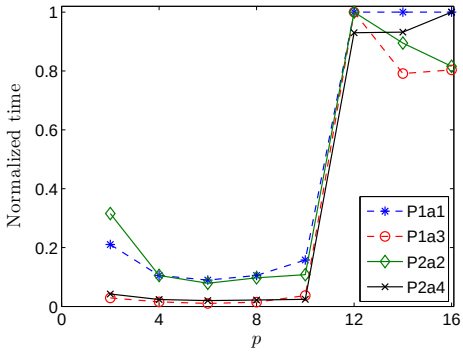
(a) $\{\beta, m, p\} = \{0.2, 10, 6\}$



(b) $\{\omega, m, p\} = \{0.2, 10, 6\}$



(c) $\{\omega, \beta, p\} = \{0.2, 0.2, 6\}$



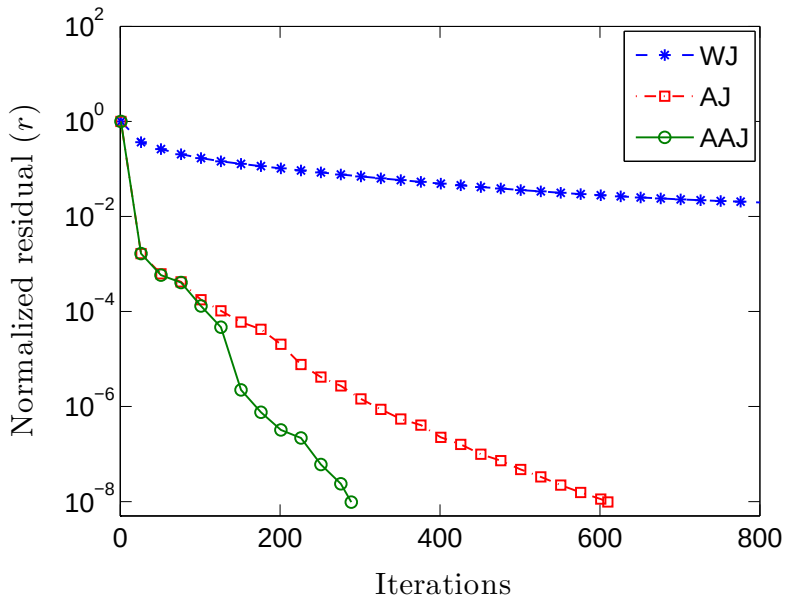
(d) $\{\omega, \beta, m\} = \{0.2, 0.2, 10\}$

Figure 7: Performance of AAJ for different choices of parameters. The computational times within any curve are normalized with respect to the maximum value in that curve. The linear systems are obtained from the discretization of the one-dimensional Laplace equation with zero Dirichlet and Neumann boundary conditions.

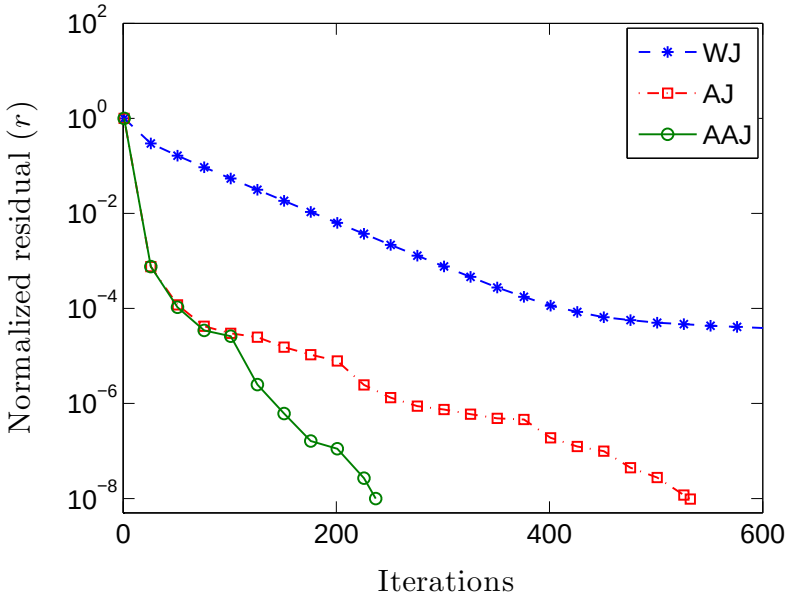
Next, we compare in Fig. 8 the progression of the relative residual during the WJ, AJ, and AAJ fixed-point iterations. We observe that the AAJ method converges extremely rapidly, while maintaining a relatively high rate of convergence throughout the iteration. In fact, AAJ is able to reduce the normalized residual to 1×10^{-8} in 107

and 72 times fewer iterations compared to WJ for the systems 'P1a1' and 'P2a1', respectively. Remarkably, the AAJ technique also requires fewer iterations to achieve a specified tolerance compared to the AJ method. This suggests that iterates produced by WJ updates are better suited for Anderson extrapolation than those produced by AJ updates. In practice, we find that the WJ iterations effectively reduce 'higher-frequency' components of the error while Anderson extrapolations effectively reduce 'lower-frequency' components, yielding a combined method effective at reducing both.

Finally, we present in Fig. 9 the speed-ups of the AJ and AAJ methods relative to WJ as a function of system size. It is clear that both AJ and AAJ are able to significantly accelerate the convergence of the WJ method. In fact, AAJ is able to achieve staggering speed-ups in excess of $19,000$ and 100 for the largest systems in the Dirichlet and Neumann problems, respectively. At the same time, AAJ is able to accelerate AJ by up to an order of magnitude. Notably, the trends in the plots indicate that even larger speed-ups of AJ and AAJ over WJ are expected as n_d is increased. Overall, we conclude that AAJ is not only able to tremendously accelerate the WJ method, but also able to noticeably outperform AJ as well.

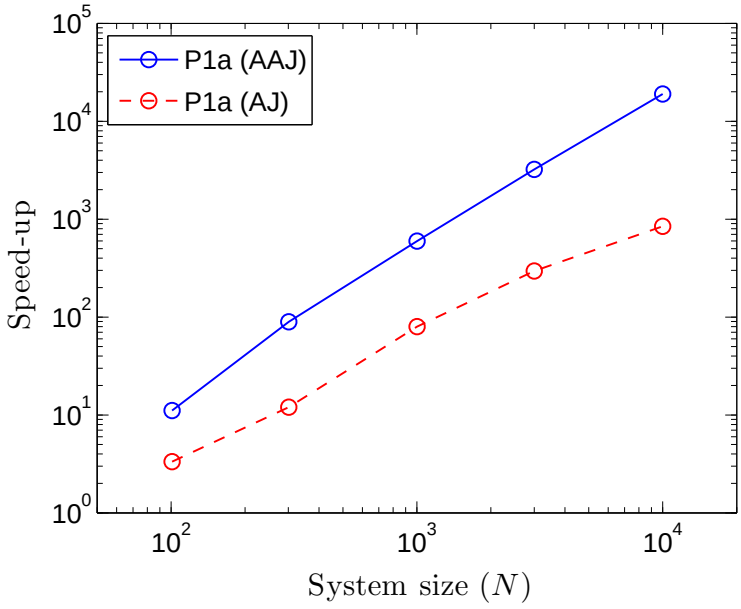


(a) P1a1

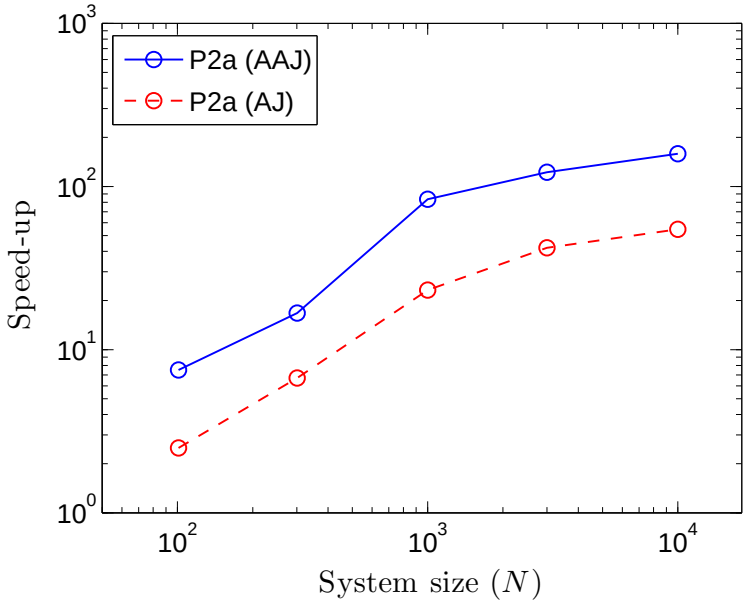


(b) P2a1

Figure 8: Comparison of the convergence of the WJ, AJ, and AAJ methods. The linear systems are obtained from the discretization of the one-dimensional Laplace equation with zero Dirichlet and Neumann boundary conditions.



(a) Problem 1



(b) Problem 2

Figure 9: Speed-up of AJ and AAJ methods relative to WJ. The linear systems are obtained from the discretization of the one-dimensional Laplace equation with zero Dirichlet and Neumann boundary conditions.

Two-dimensional Laplace equation The WJ method presented in Section 3.6.1 employs a constant relaxation parameter ω. However, this condition can be relaxed to accelerate the WJ method, as demonstrated by the recently developed Scheduled Relaxation Jacobi (SRJ) method [118]. In order to facilitate comparison with SRJ, we consider the two-dimensional Laplace equation with zero Neumann boundary conditions:

$$\text{Problem 3:} \qquad -V_{xx}(x,y) - V_{yy}(x,y) = 0 \ \text{ in } \ \Omega\,, \quad \frac{\partial V(x,y)}{\partial \mathbf{n}} = 0 \ \text{ on } \ \partial\Omega\,, \tag{94}$$

where $\Omega \in \mathbb{R}^2$ is a square with side L. Specifically, we choose $L = 100$ and $n_d = 32, 64, 128,$ and 256 finite-difference nodes in each direction. The resulting systems are symmetric positive-semidefinite with solution $\mathbf{x}^* = \mathbf{c}$.

In Fig. 10, we compare the residual as a function of iteration number for the WJ, AJ, AAJ, and SRJ methods. On one hand, SRJ demonstrates significantly larger asymptotic convergence rates compared to WJ. Therefore, in situations where high accuracies are desired, the SRJ method may be expected to significantly outperform the WJ method. On the other hand, WJ quickly reduces the initial residual compared to SRJ, which follows from its ability to rapidly nullify the 'high frequency' components of the error/residual. We find both AJ and AAJ methods require fewer iterations than SRJ, with AAJ demonstrating the most rapid convergence of all.

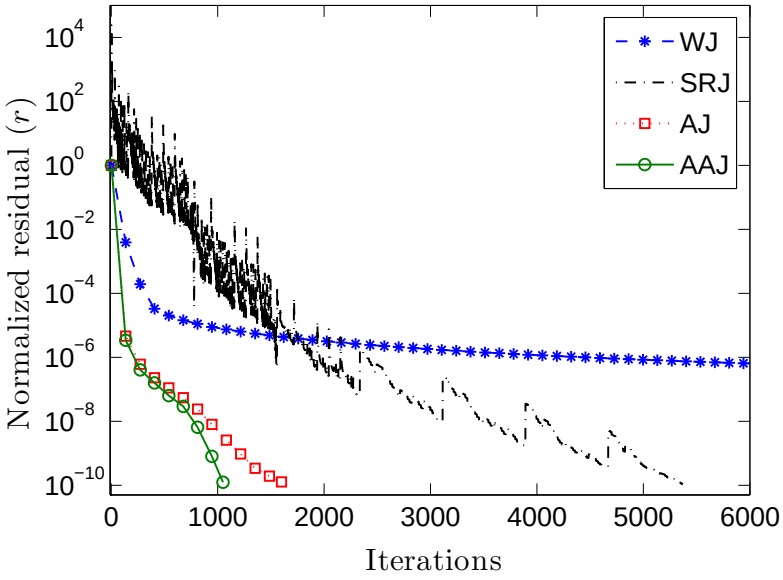


Figure 10: Convergence of WJ, AJ, AAJ, and SRJ methods for the 'P3a4' system. The linear systems are obtained from discretization of the two-dimensional Laplace equation with zero Neumann boundary conditions.

In Table 3, we compare the ability of the AAJ and SRJ methods to accelerate the WJ method. We observe that AAJ demonstrates larger speed-ups compared to SRJ. Most notably, the performance of AAJ relative to SRJ improves with size of the system.

Table 3: Speed-up of AAJ and SRJ methods relative to WJ. The linear systems are obtained by discretization of the two-dimensional Laplace equation with zero Neumann boundary conditions.

	P3a1	P3a2	P3a3	P3a4
AAJ	4.0	14.3	44.1	61.0
SRJ	5.0	10.8	21.1	26.3

3.6.4.2 Electronic structure calculations: Poisson and Helmholtz equations

We now focus on linear systems arising from the discretization of partial differential equations arising in electronic structure calculations. Specifically, we consider the Poisson and complex-valued Helmholtz equations discretized using sixth-order finite differences. As for the Laplace problems, we employ the optimal relaxation parameter ω^* for WJ, $\{\beta, m\} = \{0.2, 10\}$ for AJ, and $\{\omega, \beta, m, p\} = \{0.2, 0.2, 10, 6\}$ for AAJ. When comparing with the Krylov subspace method GMRES (restarted every 30 iterations), we calculate the relative residual within the WJ, AJ, and AAJ methods using the relation

$$r_{k+1} = \frac{\|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|}{\|\mathbf{b}\|} = \frac{\|\mathbf{D}^{-1}\mathbf{A}\,\mathbf{x}_k - \mathbf{D}^{-1}\mathbf{b}\|}{\|\mathbf{D}^{-1}\mathbf{b}\|} = \frac{\|\mathbf{f}(\mathbf{x}_k)\|}{\|\mathbf{D}^{-1}\mathbf{b}\|}\,. \tag{95}$$

Above, the second equality follows from the use of the finite-difference approximation with a uniform mesh, whereby the diagonal elements of $\mathbf{A}$ (and therefore $\mathbf{D}$) have the same value. Another implication of this property is that the performance of GMRES for the systems $\mathbf{D}^{-1}\mathbf{A}\mathbf{x} = \mathbf{D}^{-1}\mathbf{b}$ and $\mathbf{A}\mathbf{x} = \mathbf{b}$ is identical for the discretized problems considered in this section. Unless specified otherwise, we utilize a vector of all ones as the starting guess $\mathbf{x}_0$ and $\epsilon = 1 \times 10^{-8}$ as the tolerance for convergence.

Poisson equation We now consider the three-dimensional non-periodic and periodic Poisson equations arising in real-space Density Functional Theory (DFT) [77,

106, 104, 78] and orbital-free Density Functional Theory (OF-DFT) [33, 107] simulations:

$$\text{Problem 4:} \quad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) = \rho(\mathbf{r}) + b(\mathbf{r}) \ \text{ in } \ \Omega, \quad \begin{cases} V(\mathbf{r}) = 0 \ \text{ on } \ \partial\Omega\,, \end{cases} \tag{96}$$

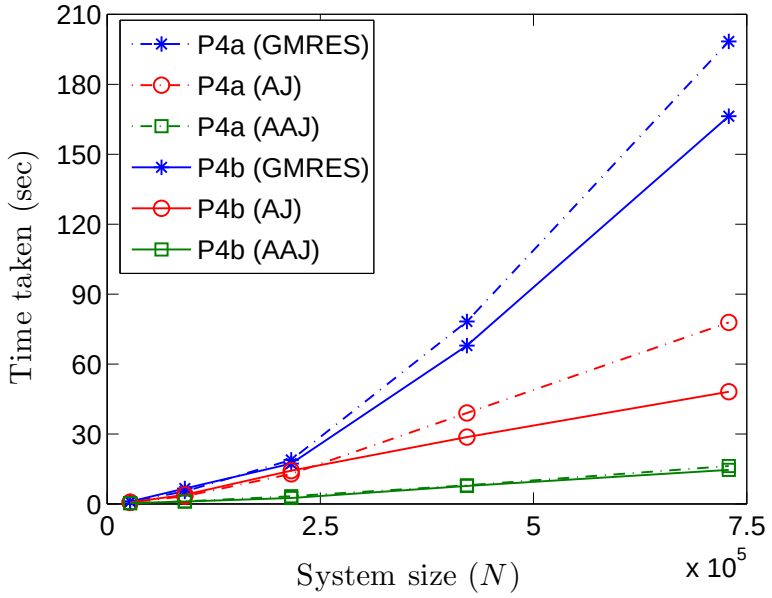
$$\text{Problem 5:} \quad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) = \rho(\mathbf{r}) + b(\mathbf{r}) \ \text{ in } \ \Omega, \quad \begin{cases} V(\mathbf{r}) = V(\mathbf{r} + L\hat{\mathbf{e}}_i) \ \text{ on } \ \partial\Omega\,, \\ \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r}) = \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r} + L\hat{\mathbf{e}}_i) \\ \text{on } \ \partial\Omega\,. \end{cases} \tag{97}$$

Above, $\Omega \in \mathbb{R}^3$ is a cubic domain of side L and $\hat{\mathbf{e}}_i$ are the unit vectors aligned with the edges of Ω (same as that in sub-section 3.5.1.1). The fields $\rho(\mathbf{r})$ and $b(\mathbf{r})$ denote the electron density and nuclear density, respectively. $\rho(\mathbf{r})$ is calculated by superimposing isolated-atom electron densities. Similarly, $b(\mathbf{r})$ is evaluated by superimposing the charge densities calculated from the highest occupied angular momentum component of the Troullier-Martins pseudopotential [111] using the finite-difference approximation [105, 107]. In Table 4, we present the nomenclature and details for the various systems of equations corresponding to the aforementioned problems.

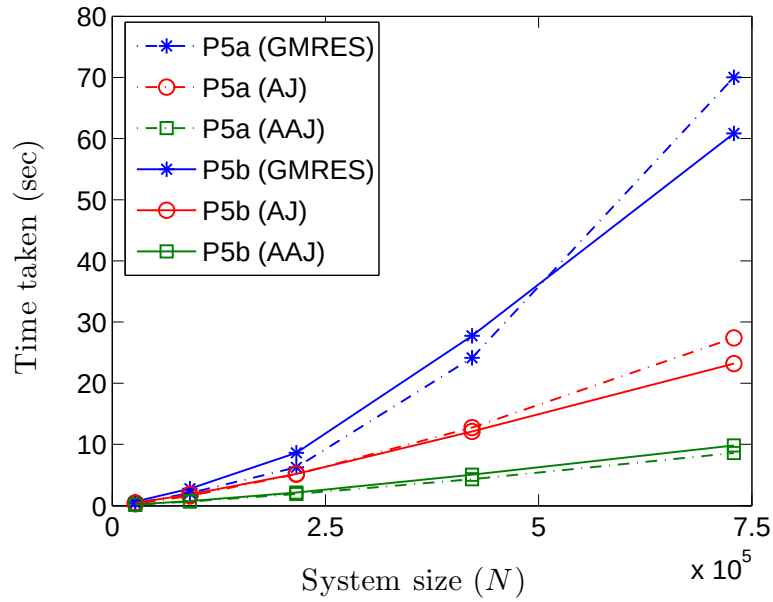
Table 4: Nomenclature for the different linear systems arising from the discretization of the three-dimensional non-periodic and periodic Poisson equations. 'P4a' corresponds to a Si_5H_{12} cluster with varying h, whereas 'P4b1', 'P4b2', 'P4b3', 'P4b4', and 'P4b5' correspond to Si_5H_{12}, $Si_{17}H_{36}$, $Si_{87}H_{76}$, $Si_{275}H_{172}$, and $Si_{525}H_{276}$ clusters, respectively. 'P5a' denotes varying h for a single diamond cubic unit cell of Silicon, whereas 'P5b1', 'P5b2', 'P5b3', 'P5b4', and 'P5b5' correspond to 2, 3, 4, 5, and 6 diamond cubic unit cells of Silicon in each direction with a vacancy. The lattice constant of diamond cubic Silicon is chosen to be 10.26 Bohr.

Discretization	Nodes in each direction (n_d)				
parameters	30	45	60	75	90
$L = 28.50$ Bohr	P4a1	P4a2	P4a3	P4a4	P4a5
$h = 0.98$ Bohr	P4b1	P4b2	P4b3	P4b4	P4b5
$L = 10.26$ Bohr	P5a1	P5a2	P5a3	P5a4	P5a5
$h = 0.68$ Bohr	P5b1	P5b2	P5b3	P5b4	P5b5

In Fig. 11, we compare the performance of the AJ, AAJ, and GMRES methods by plotting the time taken as a function of system size. We observe that both AAJ and AJ are able to outperform GMRES, with AAJ comfortably demonstrating the best timings. In particular, AAJ exhibits close to linear scaling with system size, making it an attractive technique for solving large systems of equations. Notably, AAJ achieves a speed-up of nearly an order of magnitude over GMRES for systems of size $N = 729,000$, with the speed-up increasing as the system gets larger. This is indeed verified by the results in Table 5. Significantly, for the 'P4b7' system, AAJ is faster than GMRES and WJ by factors in excess of 20 and 100, respectively.



(a) Problem 4



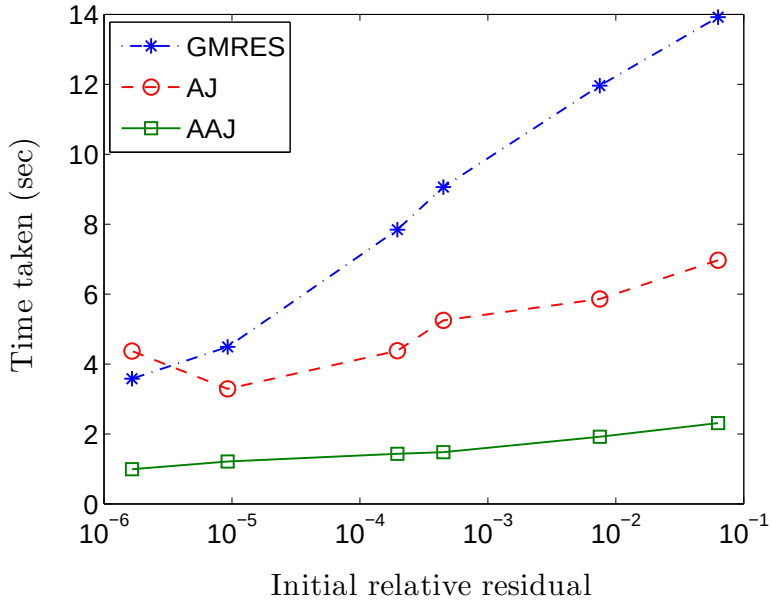
(b) Problem 5

Figure 11: Performance of AJ, AAJ, and GMRES methods for linear systems obtained from the discretization of the three-dimensional non-periodic and periodic Poisson equations.

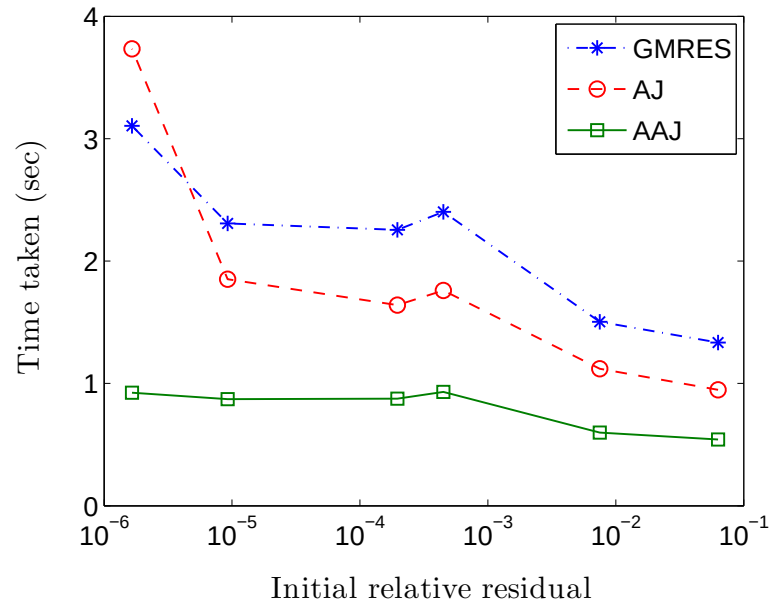
Table 5: Computational time in seconds taken by the WJ, AJ, AAJ, and GMRES approaches. 'P4b6' and 'P4b7' correspond to the $Si_{525}H_{276}$ cluster whereas 'P5b6' and 'P5b7' correspond to 6 diamond cubic unit cells of Silicon in each direction with a vacancy. In the label, the last characters of 6 and 7 correspond to $n_d = 120$ and $n_d = 150$, respectively. The linear systems are obtained from the discretization of the three-dimensional non-periodic and periodic Poisson equations.

Method	Problem 4		Problem 5	
	P4b6	P4b7	P5b6	P5b7
WJ	3275.10	10030.40	1342.02	4157.14
AJ	331.63	766.45	87.61	227.52
AAJ	39.60	97.67	34.18	85.03
GMRES	682.93	2038.26	311.09	908.75

Next, we study the influence of the quality of the initial guess on the performance of AAJ, for which we choose 'P4a3' as the representative example. Using an in-house code, we perform DFT calculations using the Anderson mixing accelerated SCF method, wherein Eqn. 96 is solved once every SCF iteration. As the iteration progresses towards convergence, the quality of the guess improves by virtue of using the previous step's solution. In Fig. 12, we compare the performance of the AAJ method with the AJ and GMRES techniques for different initial relative residuals. Specifically, we plot in Fig. 12a the time taken for the relative residual to reach $\epsilon = 1 \times 10^{-8}$. We also plot in Fig. 12b the time taken to reduce the relative residual by a factor of 1×10^{-2}. We find that AAJ significantly outperforms AJ and GMRES irrespective of the quality of the initial guess. In particular, the performance of AAJ is relatively independent of the nature of the initial guess.



(a) Convergence to a tolerance of $\epsilon = 1 \times 10^{-8}$.



(b) Initial relative residual reduction by a factor of 0.01.

Figure 12: Performance of AJ, AAJ, and GMRES methods as a function of the quality of the initial guess. The linear system under consideration is 'P4a3', obtained by the discretization of the three-dimensional non-periodic Poisson equation.

Interestingly, the time taken by GMRES and AJ to reduce the relative residual by two orders of magnitude increases as the initial guess gets closer to the converged solution. Based on these observations, we can surmise that the efficiency of AAJ can be partly attributed to its enhanced performance as the relative residual becomes smaller during the linear solve.

Helmholtz equation Next, we consider the Helmholtz equation arising in periodic real-space OFDFT calculations [23, 33]:

$$\text{Problem 6:} -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) + Q\,V(\mathbf{r}) = P\,\rho^{\alpha}(\mathbf{r}) \;\text{ in }\; \Omega, \quad \begin{cases} V(\mathbf{r}) = V(\mathbf{r} + L\hat{\mathbf{e}}_i) \;\text{ on }\; \partial\Omega\,, \\ \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r}) = \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r} + L\hat{\mathbf{e}}_i) \\ \text{on }\; \partial\Omega\,, \end{cases} \tag{98}$$

where $\Omega \in \mathbb{R}^3$ is a cubic domain of side L. The constants $\alpha = \frac{5}{6} + \frac{\sqrt{5}}{6}$, $P = 0.0296 + i\,0.0217$, and $Q = -0.1284 - i\,0.1269$. The resulting matrices $\mathbf{A}$ are complex-symmetric non-Hermitian. As before, the electron density $\rho(\mathbf{r})$ is evaluated by superimposing isolated-atom electron densities. In Table 6, we present the nomenclature for the resulting systems of equations.

Table 6: Nomenclature for the different linear systems of equations arising from the discretization of the three-dimensional periodic Helmholtz equation. 'P6a' denotes a single face centered cubic (FCC) unit cell of Aluminum with varying h, whereas 'P6b1', 'P6b2', 'P6b3', 'P6b4', and 'P6b5' correspond to 2, 3, 4, 5, and 6 FCC unit cells of Aluminum in each direction, with a vacancy. The lattice constant of FCC Aluminum is chosen to be 7.65 Bohr.

Discretization	Nodes in each direction (n_d)				
parameters	30	45	60	75	90
$L = 7.65$ Bohr	P6a1	P6a2	P6a3	P6a4	P6a5
$h = 0.51$ Bohr	P6b1	P6b2	P6b3	P6b4	P6b5

In Fig. 13, we compare the performance of the AJ, AAJ, and GMRES methods for the aforedescribed linear systems of equations. It is clear that AJ and AAJ are again able to outperform GMRES, with AAJ demonstrating the best performance. Furthermore, AAJ is able to achieve close to linear scaling with system size, and therefore its performance relative to AJ and GMRES increases for larger systems. In particular, AAJ demonstrates nearly an order of magnitude speed-up over GMRES for the 'P6a5' system. It is worth noting that unlike GMRES and AJ, which show large differences in solution times for fixed-domain and fixed-mesh cases, AAJ has nearly identical performance. Overall, we conclude that AAJ represents a highly efficient method compared to Krylov subspace methods like GMRES, even for complex non-Hermitian linear systems of equations.

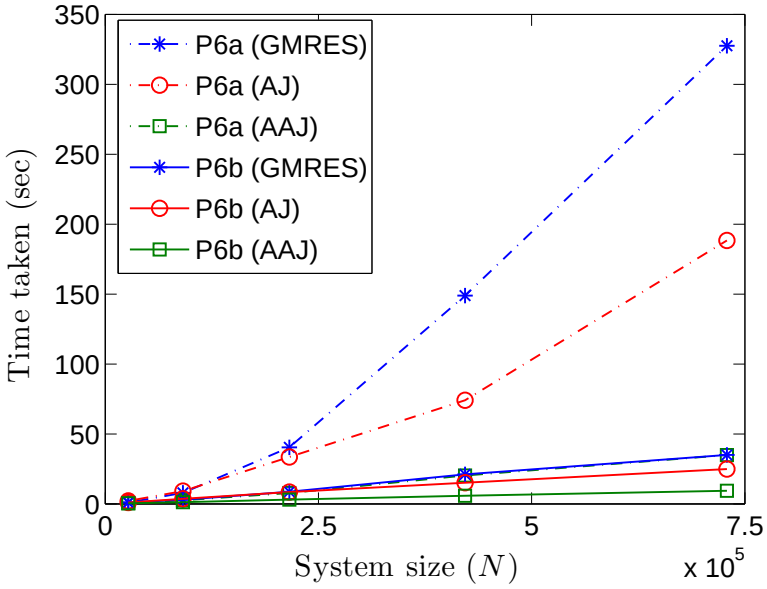


Figure 13: Performance of AJ, AAJ, and GMRES methods for the linear systems obtained from the discretization of the three-dimensional periodic Helmholtz equation.

The superior performance of AJ/AAJ compared to GMRES merits further consideration. Notably, when Anderson's extrapolation with complete history ($m = \infty$) is applied to the Richardson iteration, it is equivalent to GMRES without restart in exact arithmetic [95]. This is because GMRES and Anderson's method utilize the same Krylov subspace—albeit with a different parametrization—within which the residual is minimized. In numerical computations, non-restarted Jacobi preconditioned GMRES is expected to perform favorably compared to complete-history AJ/AAJ since linear dependency within the Krylov subspace is prevented through orthogonalization. However, restarted GMRES is almost always employed in practice to reduce orthogonalization and storage costs. Similarly, finite mixing histories are essential to the performance of AJ/AAJ. A significant difference between the GMRES restart and the finite-history AJ/AAJ used in practice is that the restart in GMRES starts the approach afresh, while the AJ/AAJ methods retain a constant mixing history size. Leveraging more such history information at each iteration may contribute to

the superior performance of AJ/AAJ over GMRES in practice. A more complete understanding of why AJ is able to outperform GMRES and why AAJ is able to outperform AJ is a worthy subject of further research.

3.6.4.3 Matrix Market: Nonsymmetric matrices

Finally, we demonstrate the generality of AAJ by considering nonsymmetric linear systems obtained via finite-element discretizations. Specifically, we consider the FIDAPM series of matrices in the Matrix Market[1] repository. In Table 7, we present the computational time taken by AAJ and GMRES for three of these systems. Within AAJ, we choose two values of the relaxation parameter $\beta = \omega$, while retaining $\{m, p\} = \{10, 6\}$ as in all the previous examples. We compare the results so obtained with GMRES for two choices of restarts: 30 and 750 iterations. In order to ensure a fair comparison, we use Jacobi preconditioning with GMRES, i.e., we solve the system $\mathbf{D}^{-1}\mathbf{A}\mathbf{x} = \mathbf{D}^{-1}\mathbf{b}$, using the same $\mathbf{D}^{-1}$ as in the Jacobi iteration. The tolerance for the relative residual is set to $\epsilon = 1 \times 10^{-8}$ and a vector of all ones is used as the starting guess $\mathbf{x}_0$. We observe that AAJ is able to outperform GMRES for these nonsymmetric finite-element matrices and choice of parameters. Overall, while applicable to nonsymmetric systems with a variety of spectra, we find that AAJ is generally less efficient for systems wherein the smallest real part of the eigenvalues of $\mathbf{D}^{-1}\mathbf{A}$ are negative, as may be expected given its Jacobi aspect. As an example, for the 'utm300' system in the TOKAMAK collection, GMRES with restart of 150 is factor of 1.2 faster than AAJ with $\{\omega, \beta, m, p\} = \{0.3, 0.3, 150, 6\}$.

[1] http://math.nist.gov/MatrixMarket/

Table 7: Time taken in seconds by AAJ and GMRES for linear systems from Matrix Market. In the table, '-' indicates that convergence was not achieved within 1000 sec.

Matrix	N	AAJ		GMRES for $\mathbf{D}^{-1}\mathbf{A}\,\mathbf{x} = \mathbf{D}^{-1}\mathbf{b}$	
		$\beta = 0.3$	$\beta = 0.4$	$restart = 30$	$restart = 750$
fidap008	3096	315.05	152.72	-	877.14
fidap029	2870	0.009	0.007	0.014	0.014
fidapm37	9152	91.76	26.40	-	233.89

CHAPTER IV

NUMERICAL IMPLEMENTATION

In this chapter, we discuss the numerical implementation of the Clenshaw-Curtis Spectral Quadrature (SQ) method to solve the DFT equations in density matrix formulation. We do so in the context of the finite-difference representation, a commonly used discretization scheme in electronic structure calculations [1, 21]. We employ the Self-Consistent Field (SCF) iteration, wherein the electron density is iterated to the ground state (fixed point), after which total energy and atomic forces are computed. In this setting, we express all quantities of interest as bilinear forms, or sums over bilinear forms, and then approximate them using Clenshaw-Curtis quadrature rules.

4.1 Preliminaries

In view of the nearsightedness principle [90], we define the *region of influence* of any point in space as the cube of side $2R_{cut}$ centered at that point. We choose a cube rather than a sphere for simplicity and efficiency within the finite-difference discretization. The parameter R_{cut} corresponds to the truncation radius of the density matrix, the distance beyond which electronic interactions can be ignored. Indeed, the magnitude of the electronic interactions decreases exponentially with distance for insulators as well as metallic systems at finite temperature. We exploit this decay to perform $\mathcal{O}(N)$ Γ-point (periodic) calculations as well as infinite-cell (infinite-crystal) calculations.

In this implementation, we assume the unit cell Ω to be cubical. We discretize the domain using a three dimensional finite-difference grid with uniform mesh size h in all directions. Let n_{proc} be the total number of processors among which the computational load is distributed. In this work, we have distributed the finite-difference nodes equally amoing all processors. Each processor P is responsible for the computation

of quantities of interest at the set of finite-difference nodes, denoted by K_{Ω_P}, in the processor-domain Ω_P associated with that processor. We denote the set of all the finite-difference nodes in Ω by K_Ω.

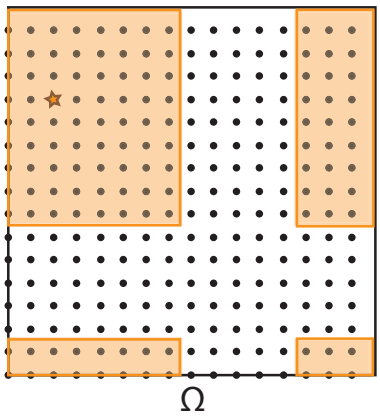
4.1.1 Domain, discretization and boundary conditions

We distinguish between Γ-point and infinite-cell calculations in the choice of domain and prescribed boundary conditions. For Γ-point calculations, we impose periodic boundary conditions on Ω, as shown in Fig. 14a. For infinite-cell calculations, we periodically extend Ω to Ω' and impose zero Dirichlet boundary conditions on Ω' for the Hamiltonian, as shown in Fig. 14b. However, we retain periodic boundary conditions on Ω for the Poisson problem. The size of Ω' is chosen such that it encompasses the region of influence of all finite-difference nodes in Ω. So, the boundary of Ω' is at a distance of R_{cut} from that of Ω. In this setting, we denote the discrete Hamiltonian, nonlocal pseudopotential matrix due to the I^{th} atom, and gradient by $\mathbf{H} \in \mathbb{R}^{N_d \times N_d}$, $\mathbf{V}_{nl}^I \in \mathbb{R}^{N_d \times N_d}$, and $\tilde{\nabla}_h \equiv (\nabla_{xh} \in \mathbb{R}^{N_d \times N_d}, \nabla_{yh} \in \mathbb{R}^{N_d \times N_d}, \nabla_{zh} \in \mathbb{R}^{N_d \times N_d})$, respectively. Here, N_d denotes the number of finite-difference nodes used to discretize Ω and Ω' in Γ-point and infinite-cell calculations, respectively, and h represents the finite-difference mesh size.

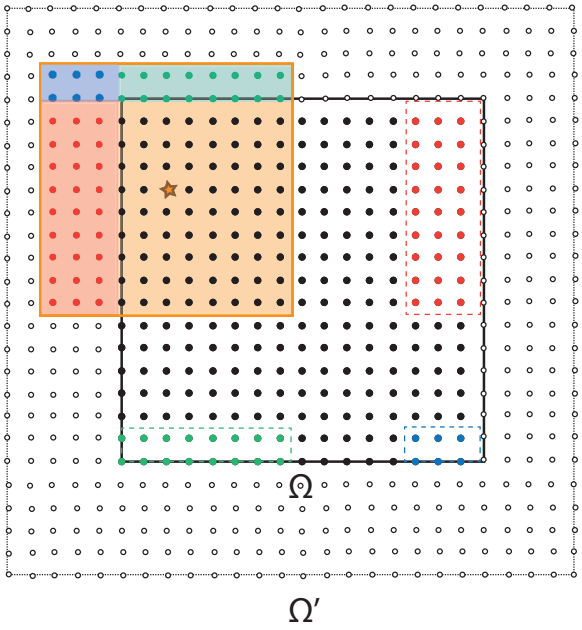
4.1.2 Finite-differences

We consider cubical domains Ω and Ω', whose edge lengths are denoted using the common notation L. We employ a uniform finite-difference grid with spacing h such that $L = n_d h$, where n_d is the number of grid points in each direction. We index the grid points by (i, j, k), where $i, j, k = 1, 2, \ldots, n_d$. We approximate the Laplacian of a function $f(\mathbf{x})$ at the grid point (i, j, k) using high-order finite-differences [62]

$$\begin{aligned} \nabla_h^2 f\big|^{(i,j,k)} \approx & \sum_{q=0}^{n_o} w_q \Big(f^{(i+q,j,k)} + f^{(i-q,j,k)} + f^{(i,j+q,k)} \\ & + f^{(i,j-q,k)} + f^{(i,j,k+q)} + f^{(i,j,k-q)} \Big), \end{aligned} \tag{99}$$



(a) Γ-point calculation



(b) Infinite-cell calculation

Figure 14: Simulation domains and discretization used for Γ-point and infinite-cell calculations. Finite-difference nodes are represented by circles, with the shaded part defining the region of influence for the finite-difference node represented by the star. Periodic and zero Dirichlet boundary conditions are prescribed on Ω and Ω', respectively.

where $f^{(i,j,k)}$ represents the value of the function $f(\mathbf{x})$ at the grid point (i,j,k). The weights are of the form [68, 50, 107]

$$\begin{aligned} w_0 &= -\frac{1}{h^2}\sum_{r=1}^{n_o}\frac{1}{r^2}\,, \\ w_q &= \frac{2(-1)^{q+1}}{h^2q^2}\frac{(n_o!)^2}{(n_o-q)!(n_o+q)!}\,,\ q=1,2,\ldots,n_o. \end{aligned} \tag{100}$$

Similarly, we approximate the gradient $\tilde{\nabla}_h \equiv (\nabla_{xh}, \nabla_{yh}, \nabla_{zh})$ as

$$\begin{aligned} \nabla_{xh} f\big|^{(i,j,k)} &= \sum_{q=1}^{n_o}\tilde{w}_q\left(f^{(i+q,j,k)} - f^{(i-q,j,k)}\right)\,, \\ \nabla_{yh} f\big|^{(i,j,k)} &= \sum_{q=1}^{n_o}\tilde{w}_q\left(f^{(i,j+q,k)} - f^{(i,j-q,k)}\right)\,, \\ \nabla_{zh} f\big|^{(i,j,k)} &= \sum_{q=1}^{n_o}\tilde{w}_q\left(f^{(i,j,k+q)} - f^{(i,j,k-q)}\right)\,, \end{aligned} \tag{101}$$

where the weights [68, 50, 107]

$$\tilde{w}_q = \frac{(-1)^{q+1}}{hq}\frac{(n_o!)^2}{(n_o-q)!(n_o+q)!}\,,\ q=1,2,\ldots,n_o. \tag{102}$$

These finite-difference expressions for the Laplacian and gradient represent $2n_o$-order accurate approximations, i.e., error $\mathcal{O}(h^{2n_o})$.

4.1.3 Integrals and summations

While performing spatial integrations, we assume that the function $f(\mathbf{x})$ is constant in a cube of side h around each grid point. For example,

$$\int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx h^3\sum_{i,j,k}^{n_d} f^{(i,j,k)} = h^3\sum_{p\in K_\Omega} f(\mathbf{x}_p) = h^3\sum_{P=1}^{n_{proc}}\sum_{p\in K_{\Omega_P}} f_p. \tag{103}$$

where the value of the function f at a point $\mathbf{x}_p \in \Omega$ is represented as f_p. For periodic integrands, this is equivalent to a trapezoidal rule. Using this rule, we approximate the nonlocal pseudopotential operator in Γ-point calculations as

$$\mathcal{V}_{\mathrm{nl}} f\big|^{(i,j,k)} = \sum_I \mathcal{V}_{\mathrm{nl}}^I f\big|^{(i,j,k)} \approx h^3\sum_I\sum_{lm}\sum_{p,q,r}^{n_d}\gamma_l^I\,\chi_{lm}^{I(i,j,k)}\chi_{lm}^{I(p,q,r)}f^{(p,q,r)}\,, \tag{104}$$

where the summation index I runs over all atoms in Ω. Analogously, for infinite-cell calculations

$$\mathcal{V}_{\text{nl}}^{\infty} f\big|^{(i,j,k)} = \sum_{J} \mathcal{V}_{\text{nl}}^{J} f\big|^{(i,j,k)} \approx h^3 \sum_{J} \sum_{lm} \sum_{p,q,r}^{n_d} \gamma_l^J \chi_{lm}^{J(i,j,k)} \chi_{lm}^{J(p,q,r)} f^{(p,q,r)} \,. \tag{105}$$

where the summation index J runs over all atoms in Ω as well as their periodic images. Since the projectors of each atom are localized in real-space, the nonlocal pseudopotential matrix can be created in $\mathcal{O}(N)$ fashion. We enforce periodic boundary conditions on Ω by employing the following strategy. In the finite-difference representations of the Laplacian, gradient, and the nonlocal pseudopotential as presented in Eqns. 99, 101 and 104 respectively, we map any index that does not correspond to a node in the finite-difference grid in Ω to its periodic image within Ω. Similarly, we enforce zero Dirichlet boundary conditions on Ω' by setting $f^{(i,j,k)} = 0$ for any index that does not correspond to a node in the finite-difference grid.

We generate the initial electron density for the Self-Consistent Field (SCF) iteration by superposing isolated-atom electron densities. We do so by visiting only atoms whose isolated-atom electron densities have nonzero overlap with Ω. Similarly, we calculate the charge density of the nuclei using the relations

$$b^{(i,j,k)} = \sum_{J} b_J^{(i,j,k)} \,, \quad b_J^{(i,j,k)} = -\frac{1}{4\pi} \nabla^2 V_J \big|^{(i,j,k)} \,, \tag{106}$$

where the summation reduces in practice to all atoms whose charge density has nonzero overlap with Ω. The localized nature of the above operations ensures that the evaluation of $b^{(i,j,k)}$ for all grid points scales as $\mathcal{O}(N)$.

4.1.4 Nodal quantities

K_Ω denotes the collection of finite-difference nodes used to discretize Ω. The nodal Hamiltonian $\mathbf{H}_p \in \mathbb{R}^{N_s \times N_s}$ of any node $p \in K_\Omega$ is defined as the restriction of the Hamiltonian to its region of influence:

$$\mathbf{H}_p = \mathbf{P}^T \mathbf{H} \mathbf{P} \,, \tag{107}$$

where the matrix

$$\mathbf{P} = \left[\mathbf{v}_{p_1}, \mathbf{v}_{p_2}, \ldots, \mathbf{v}_{p_{N_s}}\right] \in \mathbb{R}^{N_d \times N_s} . \tag{108}$$

Above, $\{\mathbf{v}_q\}_{q=1}^{N_d}$ denotes the standard basis of $\mathbb{R}^{N_d}$, and $\{p_1, p_2, \ldots, p_{N_s}\}$ are the finite difference nodes that lie within the region of influence of the $p \in K_\Omega$ node. Similarly, the nodal nonlocal pseudopotential matrix due to the J^{th} atom $\mathbf{V}_{nl,p}^J \in \mathbb{R}^{N_s \times N_s}$, and the nodal gradient $\tilde{\nabla}_{h,p} \equiv (\nabla_{xh,p} \in \mathbb{R}^{N_s \times N_s}, \nabla_{yh,p} \in \mathbb{R}^{N_s \times N_s}, \nabla_{zh,p} \in \mathbb{R}^{N_s \times N_s})$ are defined as

$$\mathbf{V}_{nl,p}^J = \mathbf{P}^T \mathbf{V}_{nl}^J \mathbf{P} , \tag{109}$$

$$\tilde{\nabla}_{h,p} = \mathbf{P}^T \tilde{\nabla}_h \mathbf{P} . \tag{110}$$

Analogously, $\mathbf{w}_p \in \mathbb{R}^{N_s \times 1}$ represents the restriction of the basis vector $\mathbf{v}_p$ to the region of influence of the $p \in K_\Omega$ node, i.e.,

$$\mathbf{w}_p = \mathbf{P}^T \mathbf{v}_p . \tag{111}$$

4.1.5 Chebyshev interpolation

In the Clenshaw-Curtis SQ method, functions of the nodal Hamiltonian are approximated in the Chebyshev polynomial basis. Specifically, the Chebyshev expansion of order n_{pl} for any function $f : [-1, 1] \to \mathbb{R}$ is of the form

$$f(\hat{\mathbf{H}}_p) = \sum_{j=0}^{n_{pl}} f_j T_j(\hat{\mathbf{H}}_p) , \tag{112}$$

where T_j denote the Chebyshev polynomials of degree j, and

$$\hat{\mathbf{H}}_p = (\mathbf{H}_p - \chi_p \mathbf{I})/\zeta_p , \tag{113}$$

is the scaled and shifted nodal Hamiltonian whose spectrum lies in the interval $[-1, 1]$. Here, $\mathbf{I} \in \mathbb{R}^{N_s \times N_s}$ denotes the identity matrix, $\chi_p = (\lambda_p^{max} + \lambda_p^{min})/2$ and $\zeta_p = (\lambda_p^{max} - \lambda_p^{min})/2$, with λ_p^{max} and λ_p^{min} denoting the maximum and minimum eigenvalues

of $\mathbf{H}_p$, respectively. The expansion coefficients in Eqn. 112 can be evaluated using the relation

$$f_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(r)T_j(r)}{\sqrt{1-r^2}}\,\mathrm{dr}\,, \quad j = 0, \ldots, n_{pl}\,, \tag{114}$$

where f_0 is further scaled by a factor of half. The column of $T_j(\hat{\mathbf{H}}_p)$ corresponding to the $p \in K_\Omega$ node, represented by $\mathbf{t}_p^j \in \mathbb{R}^{N_s \times 1}$, can be determined using the three term recurrence relation for Chebyshev polynomials:

$$\begin{aligned} \mathbf{t}_p^{j+1} &= 2\hat{\mathbf{H}}_p\mathbf{t}_p^j - \mathbf{t}_p^{j-1}\,, \\ \mathbf{t}_p^1 &= \hat{\mathbf{H}}_p\mathbf{w}_p\,,\ \mathbf{t}_p^0 = \mathbf{w}_p\,, \end{aligned} \tag{115}$$

with

$$\rho_p^j = \mathbf{w}_p^T\mathbf{t}_p^j \tag{116}$$

denoting the corresponding diagonal element. In the large scale parallel implementation, we use the relations between Chebyshev components, to compute them in an efficient manner. The expense is reduced by a factor of two when we use the relation $2T_m(r)T_n(r) = T_{m+n}(r) + T_{m-n}(r)$, where $m > n$ are integers.

4.2 Effective potential

The effective potential at any grid point $p \in K_\Omega$, given by $V_{\text{eff}}(\mathbf{x}_p) = V_{xc,p} + \phi_p$, is the sum of the exchange-correlation potential which depends only on the electron density at that point ($V_{xc,p} = V_{xc}(\rho_p)$) and the electrostatic potential which is obtained by solving the discretized Poisson's equation. In every iteration of the SCF method, we solve the discretized Poisson's equation (Eqn. 32) subject to periodic boundary conditions on Ω using the Alternating Anderson Jacobi (AAJ) method [88] (Section 3.6). Since the solution so obtained is unique only up to an arbitrary constant, we enforce the condition $\phi^{(1,1,1)} = 0$ for definiteness. In every subsequent Poisson equation encountered, we use the previous solution as starting guess. We note that sophisticated preconditioners such as multigrid [44] must be employed for

the Poisson equation in order to achieve $\mathcal{O}(N)$ scaling in practice. However, since the Poisson solve constitutes a small fraction of the total computation in our current serial implementation, such preconditioning schemes have not been employed in the present work. AAJ has shown efficient performance and is better suited for massive parallelization as it has a favorable prefactor and scaling (Section 3.6) and minimal global communication among processors [108]. We note that communication between neighboring processors is required to share the information on effective potential as discussed in sub-section 4.6.2.

4.3 Potential mixing

Non-linear as well as linear fixed-point iterations can be accelerated using mixing techniques (Chapter 3) such as Anderson's extrapolation [4], restarted-Pulay mixing [86] or periodic-Pulay mixing [8]. Pulay mixing [91] is one of the most widely used acceleration techniques for SCF iterations. Both restarted-Pulay as well as periodic-Pulay techniques have shown robust and efficient performance compared to the standard Anderson/Pulay mixing technique. In this work, we employ the periodic-Pulay mixing to accelerate the SCF iteration. Periodic-Pulay mixing is similar in spirit to the AAJ linear solver technique and hence is amenable to massive parallel calculations. In this work, we perform potential mixing where the iterates that are used come up with an improved update are the discretized effective potential functions evaluated in each SCF iteration. In periodic-Pulay mixing, a simple mixing update depending only on the previous two iterates is performed in every SCF iteration and the update is periodically obtained using Anderson extrapolation scheme as a weighted sum of the previous iterates that minimizes the 2-norm of the residual of the fixed-point problem.

4.4 Electron density

The electron density (Eqn. 23) needs to be evaluated in each iteration of the SCF method. In order to achieve this in $\mathcal{O}(N)$ fashion, we first utilize the exponential decay in the density matrix to express the electron density at the $p \in K_\Omega$ finite-difference node as

$$\rho_p = \frac{2}{h^3}\mathbf{v}_p^T g(\mathbf{H},\mu,\sigma)\mathbf{v}_p \approx \frac{2}{h^3}\mathbf{w}_p^T g(\mathbf{H}_p,\mu,\sigma)\mathbf{w}_p = \frac{2}{h^3}\mathbf{w}_p^T g(\hat{\mathbf{H}}_p,\hat{\mu}_p,\hat{\sigma}_p)\mathbf{w}_p\,, \tag{117}$$

where $\hat{\sigma}_p = \sigma/\zeta_p$ denotes the scaled smearing, and $\hat{\mu}_p = (\mu - \chi_p)/\zeta_p$ denotes the scaled and shifted Fermi energy. Next, we approximate the Fermi-Dirac function in a Chebyshev polynomial basis to arrive at

$$\rho_p \approx \frac{2}{h^3}\mathbf{w}_p^T\left(\sum_{j=0}^{n_{pl}} c_p^j(\mu)T_j(\hat{\mathbf{H}}_p)\right)\mathbf{w}_p = \frac{2}{h^3}\mathbf{w}_p^T\left(\sum_{j=0}^{n_{pl}} c_p^j(\mu)t_p^j\right) = \frac{2}{h^3}\sum_{j=0}^{n_{pl}} c_p^j(\mu)\rho_p^j\,, \tag{118}$$

where ρ_p^j are determined using Eqns. 115 and 116. The electron density at the $p \in K_\Omega$ node can therefore be expressed as

$$\rho_p = \frac{2}{h^3}\sum_{j=0}^{n_{pl}} c_p^j\rho_p^j\,, \tag{119}$$

where the expansion coefficients c_p^j correspond to the Fermi energy μ that results in the correct number of total electrons, i.e.,

$$h^3\sum_{p\in K_\Omega}\rho_p = 2\sum_{P=1}^{n_{proc}}\sum_{p\in K_{\Omega_P}}\sum_{j=0}^{n_{pl}} c_p^j(\mu)\rho_p^j = N_\Omega\,. \tag{120}$$

The above equation can be iteratively solved for μ using a root-finding algorithm such as Newton's [72] or Brent's method [15]. During this process, the coefficients c_p^j are computed using Eqn. 114 by setting $f(r) = g(r,\hat{\mu}_p,\hat{\sigma}_p)$. The above expressions are applicable to both Γ-point and infinite-cell calculations, with the nodal quantities appropriately defined as described in Subsection 4.1.4.

We note the significant differences in the density calculation in Clenshaw-Curtis SQ and classical Chebyshev polynomial FOE. In the FOE approach, the complete

density matrix is computed, whereas only its diagonal is computed in SQ. This can be seen in Eqn. 118, where the vectors $\mathbf{w}_p$ are moved inside the summation to form the scalars ρ_p^j. Indeed, such a procedure involves the calculation of the local vectors $\mathbf{t}_p^j$ using the recursive relation Eqn. 115. However, the truncated columns of the density matrix, which correspond to the local vectors $\sum_{j=0}^{n_{pl}} c_p^j \mathbf{t}_p^j$, are not computed in the SQ method. Furthermore, since the scalars ρ_p^j can be stored, the SQ approach allows for efficient determination of the Fermi energy μ, without recomputation or storage of the Chebyshev matrices, as typically done in FOE. Finally, the key operations in SQ are local matrix-vector multiplications, compared to global matrix-matrix multiplications in FOE. The evaluation of the free energy in SQ proceeds along similar lines and does not involve the calculation of the density matrix, as further described in the next section. In our parallel implementation we do not form even the local matrices, instead, we compute the action of the nodal matrix on the vector, for any given vector. This way, all the operations are local and matrix free. The matrix free approach is discussed below.

4.4.1 Matrix free approach

At each finite difference node p of processor P, we need to find the scaling parameters that can scale the nodal Hamiltonian matrix corresponding to that node such that its spectrum is in $[-1, 1]$. These parameters can be computed using Lanczos algorithm [60]. The nodal Hamiltonian matrix $\mathbf{H}_p$ corresponds to the nodes in a sub-domain around the p^{th} finite difference node and is given by,

$$\mathbf{H}_p = -\frac{1}{2}(\boldsymbol{\nabla}^2)_p + \mathbf{V}_{\text{eff},p} + \mathbf{V}_{\text{nl},p}. \tag{121}$$

Each of the matrices (bold font) in the above equation are of size $N_\text{s} \times N_\text{s}$.

From Eqns. 113 and 115, we have,

$$\mathbf{t}_p^{j+1} = \frac{2}{\zeta_p}\mathbf{H}_p\mathbf{t}_p^j - \frac{2\chi_p}{\zeta_p}\mathbf{t}_p^j - \mathbf{t}_p^{j-1} \tag{122}$$

Both Lanczos algorithm as well as computing ρ_p^j require only the product of $\mathbf{H}_p$ with a vector and hence can be carried out without explicitly forming the matrix in Eqn. 121. The elements of diagonal matrix $\mathbf{V}_{\text{eff},p}$ are computed as described in section 4.2. The $-\frac{1}{2}(\boldsymbol{\nabla}^2)_p$ matrix times a vector is computed by applying the finite difference stencil weights (Eqn. 99) at each node in the sub-domain with appropriate boundary conditions. For any given vector $\mathbf{t}_j^p$ of size $N_\text{s} \times 1$, the matrix vector product $\mathbf{H}_p\,\mathbf{t}_j^p$ can be evaluated from the products, $-\frac{1}{2}(\boldsymbol{\nabla}^2)_p\,\mathbf{t}_j^p$, $\mathbf{V}_{\text{eff},p}\,\mathbf{t}_j^p$ and $\mathbf{V}_{\text{nl},p}\,\mathbf{t}_j^p$. In the following algorithms, we discuss computation of the Chebyshev coefficients ρ_p^j, Fermi energy μ and the electron density in a matrix free way.

Algorithm 2: NODAL_NONLOCAL_TIMES_VEC($\mathbf{t}_p^j$)

Nodal Nonlocal matrix times a given vector $\mathbf{t}_p^j$ in a matrix free way.

Input: $\mathbf{t}_p^j$

Computing $\mathbf{V}_{\text{nl},p}\,\mathbf{t}_p^j$:

$A_{r_c}^P$ is the set of all atoms within a distance of $r_c + R_{cut}$ distance from Ω_P.

for $J \in A_{r_c}^P$ **do**

  For each atom, find the start (n_s) and end (n_e) nodes (in local indexing) of the intersection region within r_c distance from the sub-domain around node p.

  for $i \in \{n_s \ldots n_e\}$ **do**

    Find distance $\mathbf{r}[i]$ from each node to the atom J.

  for $l = 0$ *to* l_{max} **do**

    for $m = -l$ *to* l **do**

      $\mathbf{v} = \mathbf{0}$

      for $i \in \{n_s \ldots n_e\}$ **do**

        $\mathbf{v}[i] = \mathbf{v}[i] + \chi_{lm}^J[i]\mathbf{t}_p^j[i]$

      for $i \in \{n_s \ldots n_e\}$ **do**

        $\mathbf{t}_d[i] = \mathbf{t}_d[i] + \chi_{lm}^J[i]\mathbf{v}[i]$

Note: The set of atoms $A_{r_c}^P$ and the nonlocal projectors χ_{lm}^J are pre-computed and need not be re-calculated in every SCF iteration.

Output: $\mathbf{t}_d$

Algorithm 3: NODAL_HAMILTONIAN_TIMES_VEC($\mathbf{t}_p^j$)

Nodal Hamiltonian times a given vector $\mathbf{t}_p^j$ in a matrix free way.

Input: $\mathbf{t}_p^j$

Computing $-\frac{1}{2}(\boldsymbol{\nabla}^2)^p\,\mathbf{t}_p^j$:
for $i = 1 \ldots N_s$ **do**
  for $q = 0 \ldots 2n_o$ **do**
    Find $\tilde{q}$ as the index of the node corresponding to q.
    $\mathbf{t}_d[i] = \mathbf{t}_d[i] - \frac{1}{2} s_q \mathbf{t}_p^j[\tilde{q}]$ (where, s_q are the finite difference weights.)

Computing $\mathbf{V}_{\text{eff},p}\,\mathbf{t}_p^j$:
for $i = 1 \ldots N_s$ **do**
  $\mathbf{t}_d[i] = \mathbf{t}_d[i] + \mathbf{V}_{\text{eff},p}[i]\,\mathbf{t}_p^j[i]$

Computing $\mathbf{V}_{\text{nl},p}\,\mathbf{t}_p^j$:
$\mathbf{t}_d = \mathbf{t}_d +$ NODAL_NONLOCAL_TIMES_VEC($\mathbf{t}_p^j$)

Output: $\mathbf{t}_d$

Algorithm 6 which is executed by all the processors, provides a summary of the computation of the electron density using the algorithms described. In the algorithms, Ω_P denotes the computational domain associated with the processor P.

4.5 Energy and forces

4.5.1 Energy

The free energy (Eqn. 37) can be evaluated once the electronic ground-state is determined. Since the density matrix is not computed within the SQ method, the band structure energy and electronic entropy cannot be directly evaluated. In view of this, we develop expressions for the calculation of these quantities, as described below.

Algorithm 4: CHEBYSHEV_COMPONENTS

Compute Chebyshev components ρ_p^j.

Input: χ_p, ζ_p for all $p \in K_{\Omega_P}$.

for $p \in K_{\Omega_P}$ **do**
  $\mathbf{t}_1 = \mathbf{w}_p, \quad \mathbf{t}_2 = (\frac{1}{\zeta_p}\text{NODAL\_HAMILTONIAN\_TIMES\_VEC}(\mathbf{w}_p)) - \frac{\chi_p}{\zeta_p}\mathbf{w}_p$
  $\rho_p^0 = 1$
  $\rho_p^1 = \mathbf{w}_p^T \mathbf{t}_2$

  for $j = 2 \ldots n_{pl}$ **do**
    $\mathbf{t}_3 = (\frac{2}{\zeta_p}\text{NODAL\_HAMILTONIAN\_TIMES\_VEC}(\mathbf{t}_2)) - \frac{2\chi_p}{\zeta_p}\mathbf{t}_2 - \mathbf{t}_1$
    $\mathbf{t}_1 = \mathbf{t}_2$
    $\mathbf{t}_2 = \mathbf{t}_3$
    $\rho_p^j = \mathbf{w}_p^T \mathbf{t}_3$

Output: ρ_p^j for all $p \in K_{\Omega_P}$

Algorithm 5: FERMI_ENERGY

Compute Fermi energy μ using a root finding algorithm.
Input: $\rho_p^j, \chi_p, \zeta_p$ for all $p \in K_{\Omega_P}$.

STEP 1: Start with an initial guess for μ.
STEP 2: Compute Chebyshev coefficients c_p^j at each node p.
STEP 3: Evaluate the residual of Eqn. 120 using MPI_Allreduce.
STEP 4: Update μ using a root finding algorithm.
STEP 5: Repeat steps 2 to 4 until convergence.

Output: μ

Band structure energy Utilizing the same procedure and approximations as in the case of the electron density, the band structure energy takes the form

$$\begin{aligned}
E_{band} &= 2\sum_{p\in K_\Omega} \mathbf{v}_p^T \left(\mathbf{H}^* g(\mathbf{H}^*, \mu, \sigma)\right) \mathbf{v}_p \approx 2\sum_{p\in K_\Omega} \mathbf{w}_p^T \left(\mathbf{H}_p^* g(\mathbf{H}_p^*, \mu, \sigma)\right) \mathbf{w}_p \\
&= 2\sum_{p\in K_\Omega} \mathbf{w}_p^T \left[(\chi_p + \zeta_p \hat{\mathbf{H}}_p^*) g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p)\right] \mathbf{w}_p \\
&\approx 2\sum_{p\in K_\Omega} \mathbf{w}_p^T \left[\chi_p \sum_{j=0}^{n_{pl}} c_p^j T_j(\hat{\mathbf{H}}_p^*) + \zeta_p \sum_{j=0}^{n_{pl}} d_p^j T_j(\hat{\mathbf{H}}_p^*)\right] \mathbf{w}_p \qquad (123) \\
&= 2\sum_{P=1}^{n_{proc}} \sum_{p\in K_{\Omega_P}} \sum_{j=0}^{n_{pl}} (\chi_p c_p^j + \zeta_p d_p^j) \rho_p^{j*},
\end{aligned}$$

Algorithm 6: ELECTRON_DENSITY

Electron density ρ using Spectral Quadrature method.
Input: $V_{\text{eff},p}$ for all $p \in K_{\Omega_P}$.

STEP 1: Communicate $V_{\text{eff},p}$ at nodes outside Ω_P using MPI_Ineighbor_alltoallv.
STEP 2: Compute scaling parameters χ_p and ζ_p using Lanczos.
STEP 3: Compute ρ_p^j using CHEBYSHEV_COMPONENTS.
STEP 4: Compute FERMI_ENERGY (needs MPI_Allreduce) and update c_p^j.
STEP 5: Compute ρ_p using Eqn. 119.

Output: ρ_p for all $p \in K_{\Omega_P}$

where d_p^j are the coefficients of the Chebyshev expansion obtained by setting $f(r) = rg(r, \hat{\mu}_p, \hat{\sigma}_p)$ in Eqn. 114.

Electronic entropy The electronic entropy can similarly be written as

$$\begin{aligned}
S &= 2k_B \sum_{p \in K_\Omega} \mathbf{v}_p^T \left[g(\mathbf{H}^*, \mu, \sigma) \log g(\mathbf{H}^*, \mu, \sigma) \right. \\
&\quad \left. + (\mathbf{I} - g(\mathbf{H}^*, \mu, \sigma)) \log (\mathbf{I} - g(\mathbf{H}^*, \mu, \sigma)) \right] \mathbf{v}_p \\
&\approx 2k_B \sum_{p \in K_\Omega} \mathbf{w}_p^T \left[g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p) \log g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p) \right. \\
&\quad \left. + \left(\mathbf{I} - g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p) \right) \log \left(\mathbf{I} - g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p) \right) \right] \mathbf{w}_p \\
&\approx 2k_B \sum_{p \in K_\Omega} \mathbf{w}_p^T \left[\sum_{j=0}^{n_{pl}} e_p^j T_j(\hat{\mathbf{H}}_p^*) \right] \mathbf{w}_p \qquad (124) \\
&= 2k_B \sum_{P=1}^{n_{proc}} \sum_{p \in K_{\Omega_P}} \sum_{j=0}^{n_{pl}} e_p^j \rho_p^{j*},
\end{aligned}$$

where e_p^j are calculated using Eqn. 114 by setting

$$f(r) = g(r, \hat{\mu}_p, \hat{\sigma}_p) \log g(r, \hat{\mu}_p, \hat{\sigma}_p) + (1 - g(r, \hat{\mu}_p, \hat{\sigma}_p)) \log(1 - g(r, \hat{\mu}_p, \hat{\sigma}_p)).$$

Free Energy Using the band structure energy in Eqn. 123, and the electronic entropy in Eqn. 124, the free energy in Eqn. 37 can be written as

$$\begin{aligned}\mathcal{F}_0(\mathbf{R}) &= h^3 \sum_{P=1}^{n_{proc}} \sum_{p \in K_{\Omega_P}} \left(\frac{2}{h^3} \sum_{j=0}^{n_{pl}} (\chi_p c_p^j + \zeta_p d_p^j) \rho_p^{j*} + \varepsilon_{xc}(\rho_p^*)\rho_p^* + \frac{1}{2}(b_p - \rho_p^*)\phi_p^* \right. \\ &\left. -V_{xc}(\rho_p^*)\rho_p^* + \frac{1}{2}(\tilde{b}_p + b_p)V_{c,p} - \frac{1}{2}\sum_J \tilde{b}_{J,p}\tilde{V}_{J,p} + \frac{2\sigma}{h^3}\sum_{j=0}^{n_{pl}} e_p^j \rho_p^{j*} \right), \end{aligned} \quad (125)$$

where the integrals have been approximated using the trapezoidal rule (Eqn. 103). The subscript p is used to denote the value of the quantity at or corresponding to that finite-difference node. As before, the summation index J runs over all the atoms in Ω and their periodic replicas. The free energy expression in Eqn. 125 is applicable to both Γ-point and infinite cell calculations, with the nodal quantities defined appropriately as described in sub-section 4.1.4.

4.5.2 Forces

The atomic forces (Eqn. 38) — required for structural relaxations and molecular dynamics — need to be calculated once the electronic ground-state is determined. They consist of local and nonlocal components, whose expressions we present below.

Local component The local component of the atomic force takes the form

$$\begin{aligned}\mathbf{f}_I^l &= h^3 \sum_{I'} \sum_{p \in K_\Omega} \left(\tilde{\nabla}_h b_{I'}\big|_p (\phi_p^* - V_{\mathrm{loc},p}^{I'}) + \frac{1}{2}\tilde{\nabla}_h \tilde{b}_{I'}\big|_p \left(V_{c,p} - \tilde{V}_{I',p} \right) \right. \\ &+ \frac{1}{2}\tilde{\nabla}_h b_{I'}\big|_p \left(V_{c,p} + V_{\mathrm{loc},p}^{I'} \right) + \frac{1}{2}\tilde{\nabla}_h V_{c,I'}\big|_p \left(\tilde{b}_p + b_p \right) + \frac{1}{2} b_{I',p} \tilde{\nabla}_h V_{\mathrm{loc},p}^{I'}\big|_p \\ &\left. - \frac{1}{2}\tilde{b}_{I',p}\tilde{\nabla}_h \tilde{V}_{I'}\big|_p \right), \end{aligned} \quad (126)$$

where the integrals have been approximated using the trapezoidal rule (Eqn. 103). Again, the summation index I' runs over the I^{th} atom and all its periodic images.

Nonlocal component The nonlocal component of the force — as formulated in sub-section 2.1.3 — for a Γ-point calculation can be written as (see Eqn. 45)

$$\mathbf{f}_I^{nl} = 4 \sum_{p \in K_\Omega} \mathbf{v}_p^T \left(\mathbf{V}_{nl}^I \tilde{\nabla}_h g(\mathbf{H}^*, \mu, \sigma) \right) \mathbf{v}_p \,. \tag{127}$$

After approximating the above expression in terms of the local density matrix and expanding in Chebyshev polynomials, we arrive at

$$\mathbf{f}_I^{nl} \approx 4 \sum_{p \in K_\Omega} \mathbf{w}_p^T \left(\mathbf{V}_{nl,p}^I \tilde{\nabla}_{h,p} g(\hat{\mathbf{H}}_p^*, \hat{\mu}_p, \hat{\sigma}_p) \right) \mathbf{w}_p \approx 4 \sum_{p \in K_\Omega} \sum_{j=0}^{n_{pl}} c_p^j \mathbf{w}_p^T \mathbf{V}_{nl,p}^I \tilde{\nabla}_{h,p} \mathbf{t}_p^{j*} \,. \tag{128}$$

An alternative approach for determining the nonlocal component of the force involves individually evaluating the required diagonal- and off-diagonal components of the local density matrix using SQ. However, such a strategy is computationally expensive due to the large number of density matrix components required. This is overcome in Clenshaw-Curtis SQ using the observation that required diagonal- and off-diagonal density matrix components can be written in terms of $\mathbf{t}_p^{j*}$, which are available during the recursive iteration in Eqn. 115. However, in Gauss SQ, the required components of the density matrix need to be individually computed, which makes calculation of the atomic forces considerably more expensive.

Total force Using the local component of the force in Eqn. 126, and the nonlocal component of the force in Eqn. 128, the atomic force in Eqn. 38 can be written as

$$\begin{aligned} \mathbf{f}_I \;=\; & h^3 \sum_{P=1}^{n_{proc}} \sum_{p \in K_{\Omega_P}} \sum_{I'} \Bigg(\tilde{\nabla}_h b_{I'}\big|_p (\phi_p^* - V_{\mathrm{loc},p}^{I'}) + \frac{1}{2} \tilde{\nabla}_h \tilde{b}_{I'}\big|_p \left(V_{c,p} - \tilde{V}_{I',p} \right) \\ & + \frac{1}{2} \tilde{\nabla}_h b_{I'}\big|_p \left(V_{c,p} + V_{\mathrm{loc},p}^{I'} \right) + \frac{1}{2} \tilde{\nabla}_h V_{c,I'}\big|_p \left(\tilde{b}_p + b_p \right) + \frac{1}{2} b_{I',p} \tilde{\nabla}_h V_{\mathrm{loc},p}^{I'}\big|_p \\ & - \frac{1}{2} \tilde{b}_{I',p} \tilde{\nabla}_h \tilde{V}_{I'}\big|_p \Bigg) - 4 \sum_{p \in K_\Omega} \sum_{j=0}^{n_{pl}} c_p^j \mathbf{w}_p^T \mathbf{V}_{nl,p}^I \tilde{\nabla}_{h,p} \mathbf{t}_p^{j*} \,. \end{aligned} \tag{129}$$

For an infinite-cell calculation, while the local component of the force remains the same, the nonlocal component differs because of the different boundary conditions imposed on the nonlocal pseudopotential in Γ-point and infinite-cell calculations, as

can be seen from Eqns. 104 and 105 (and the discussion below them). Adopting the same procedure as for the nonlocal Γ-point force, we arrive at the following expression for total atomic force in infinite-cell calculations:

$$\begin{aligned}\mathbf{f}_I^{\infty} &= h^3 \sum_{P=1}^{n_{proc}} \sum_{p \in K_{\Omega_P}} \sum_{I'} \Bigg(\tilde{\nabla}_h b_{I'}\big|_p (\phi_p^* - V_{\mathrm{loc},p}^{I'}) + \frac{1}{2}\tilde{\nabla}_h \tilde{b}_{I'}\big|_p \left(V_{c,p} - \tilde{V}_{I',p}\right) \\ &+ \frac{1}{2}\tilde{\nabla}_h b_{I'}\big|_p \left(V_{c,p} + V_{\mathrm{loc},p}^{I'}\right) + \frac{1}{2}\tilde{\nabla}_h V_{c,I'}\big|_p \left(\tilde{b}_p + b_p\right) + \frac{1}{2} b_{I',p} \tilde{\nabla}_h V_{\mathrm{loc},p}^{I'}\big|_p \quad (130) \\ &- \frac{1}{2}\tilde{b}_{I',p}\tilde{\nabla}_h \tilde{V}_{I'}\big|_p - \frac{4}{h^3}\sum_{j=0}^{n_{pl}} c_p^j \mathbf{w}_p^T \mathbf{V}_{nl,p}^{I'} \tilde{\nabla}_{h,p} \mathbf{t}_p^{j*} \Bigg).\end{aligned}$$

In deriving the nonlocal component of the force in the above equation, since $\mathbf{t}_p^{j*}$ is not available for $p \notin K_\Omega$, it has been periodically mapped back to the corresponding $\mathbf{t}_p^{j*}$ for $p \in K_\Omega$.

4.6 Scalability and storage

In this section we discuss the linear scaling aspects, parallel scalibility and memory requirements of the matrix-free numerical implementation. We first disucss the computational complexity in terms of the number of operations and show that each component of the entire method scales linearly with system size. We then describe the processor communications that are involved in the method and how they dictate the parallel scalability. We also discuss the storage requirements and how well suited the method is for minimum memory implementations.

4.6.1 Computational complexity

Let us define the following variables. The number of SCF iterations is n_{scf}, the number of iterations in Lanczos algorithm is n_{lancz}, the number of iterations in the root finding algorithm for Fermi energy calculation is n_{fermi} and the number of iterations for Poisson solver (AAJ) is n_{poiss}. N_s is the number of grid points in a cube of side $2R_{cut}$ which is the localization region. N_{r_b} and N_{r_c} are the number of grid points in a cube of side $2r_b$ and $2r_c$ respectively. r_c is the pseudopotential cut-off distance and

r_b is the cut-off distance required for accurate computation of ionic charge density $b(\mathbf{x})$. N_w be the number of weights in the Laplacian stencil dependent on the order of finite difference approximation. N is the number of atoms in Ω. The average number of grid points per atom is given by c_1. Let the average number of atoms whose pseudopotential cut-off is within a distance of r_b from any grid point be denoted by c_2 and that within a distance of R_{cut} be denoted by c_3. Computing $b_J(\mathbf{x}_p, \mathbf{R}_J)$ at node p takes c_4 operations. We note that n_{fermi}, N_s, N_{r_b}, N_{r_c}, N_w, c_1, c_2, c_3 and c_4 are all constant for a given system and a given accuracy, even as the system size increases.

The number of iterations in Lanczos algorithm, n_{lancz}, do not change significantly with system size. The SCF iterations do increase with the size of the system (especially for metals) [37]. Hence, preconditioned techniques have to be explored such that n_{scf} will be independent of the size of the atomistic system [63]. In the current work, since we are interested in the high temperature regime, the acceleration techniques like Periodic Pulay would be suffcient to provide a favorable scaling with system size. The number of iterations in the linear solver n_{poiss} would also be independent of system size when a multigrid preconditioner is used [17]. In Table 8, we present the scaling of each of the component of the $\mathcal{O}(N)$ DFT method. The second column of the table ("Operations") gives the approximate expression for the total number of dominant computational operations performed in that part of the calculation. m_a and p_a are the parameters of the AAJ solver, where m_a is the history size and p_a is the frequency at which Anderson update is performed. l_m is the maximum angular momemtum component of any atom in the system. The operations required for computation of exchange-correlation potential and energy (which are also $\mathcal{O}(N)$) are not included in the table for the sake of simplicity.

Table 8: Computational complexity of each of the components of the $\mathcal{O}(N)$ Spectral Quadrature DFT method. In the table, $\mathcal{O}(N_{mv}) = \mathcal{O}(N_s N_w) + c_3\mathcal{O}(l_m^2 \min(N_{r_c}, N_s))$ is the scaling of the number of operations in evaluating the nodal matrix times vector in a matrix-free way.

Component	Operations
$b(\mathbf{x})$ calculation	$(c_1 c_2 c_4(\mathcal{O}(N_{r_b}) + \mathcal{O}(N_w N_{r_b}^{2/3})))N$
V_{eff} calculation	$(c_1 n_{poiss}(n_{scf}+1)(\mathcal{O}(N_w) + \mathcal{O}(m_a^2/p_a) + \mathcal{O}(m_a/p_a)))N$
Potential mixing	$(c_1 n_{scf}(\mathcal{O}(m_a^2/p_a) + \mathcal{O}(m_a/p_a)))N$
$\rho(\mathbf{x})$ calculation	$(c_1 n_{scf}((n_{lancz}+n_{pl})\mathcal{O}(N_{mv}) + n_{fermi}\mathcal{O}(n_{pl}^2)))N$
Energy	$c_1(\mathcal{O}(n_{pl}^2) + c_2\mathcal{O}(N_{r_b}))N$
Forces	$(c_1(n_{pl}\mathcal{O}(N_{mv}) + c_2 c_4(\mathcal{O}(N_{r_b}) + \mathcal{O}(N_w N_{r_b}^{2/3}))))N$

It can be seen that all the components of the method scale linearly with the number of atoms N. Among all the components, the electron density $\rho(\mathbf{x})$ calculation has a very high prefactor. The significant cost in electron density calculation using the SQ method arises from the prefactor $c_1 n_{pl}\mathcal{O}(N_{mv})(\approx c_1 n_{pl}\mathcal{O}(N_s))$ which indicates computation of many nodal matrix times vector products depending on the quadrature order n_{pl}. N_{mv} is the number of significant operations involved in computing the matrix vector product in a matrix free way and is proportional to the size of the localization region N_s which is independent of the size of the system. On each processor, the above prefactor is given by $N_{proc} n_{pl}\mathcal{O}(N_s)$, where, N_{proc} is the number of grid points per processor. So, on a large number of processors, when N_{proc} is very small, the cost of calculating electron density can be comparable to other components depending on n_{pl} and N_s. Although evaluating Chebyshev coefficients in the electron density calculation can be achieved in $\mathcal{O}(n_{pl}\log n_{pl})$ [16] effort, in this work we have employed the discrete orthogonality of the Chebyshev polynomials to compute the coefficients (which scales as $\mathcal{O}(n_{pl}^2)$), as we target high temperature applications that

require a very low order of quadrature (i.e. reduced prefactor).

4.6.2 Processor communications

An important feature of the matrix-free implementation of the $\mathcal{O}(N)$ DFT method is its massive parallelizability that could enable us to overcome the high prefactor of the method. In this sub-section we shall discuss the communications required among processors in this method and how they affect the parallel scaling of the code.

Local communication Majority of the communication involving neighboring processors comes from two parts of the method – residual calculation in the AAJ Poisson solver and communication of effective potential information at the nearby nodes of each processor. In the former case, residual calculation in each iteration of the solver requires computation of Laplacian of ϕ at each grid point in a processor using the finite difference stencil and this requires values of ϕ at points that belong to the neighboring processors. In the latter case, the nodal Hamiltonian matrix times vector product in sub-section 4.4.1 for some point $p \in K_{\Omega_P}$ requires the values of $\mathbf{V}_{\text{eff},p}[i]$ at points i which lie inside the localization region Ω_p (cube of size $2R_{cut}$ around p) but outside the processor domain Ω_P. In both the cases, each processor is required to communicate only with a few neighboring processors and hence does not hinder the parallel scalability of the system. We use separate communicator topologies for both the cases. We use MPI_Ineighbor_alltoallv [42] that has has been developed for extreme scalability of the local communications among all the processors. However, it should be noted that any sort of communication might effect the strong scaling beyond a certain number of processors.

Global communication Scaling the code to many hundreds of thousands of processors would not be practical if the global communications are significant. Global communication among processors is typically required for computing dot products of

vectors and summation over all grid points in Ω using MPI_Allreduce operations. In the $\mathcal{O}(N)$ DFT method, global communications are required in computing the 2-norm of the residuals, to check for convergence in SCF iteration as well as in AAJ iterations. They are also required to compute dot products for Anderson updates in SCF mixing and AAJ solvers. Additionally, MPI_Allreduce communication is required in every iteration of finding Fermi energy using Eqn. 120 and in computing energy from Eqn. 125 and force from Eqn. 129 or 130 . The total number of MPI_Allreduce operations in one Molecular Dynamics time step using the $\mathcal{O}(N)$ DFT method is approximately $(n_{scf}+1)\frac{n_{poiss}}{p_a} + n_{scf} + \frac{n_{scf}}{p_a} + n_{fermi} + 2$. This is a small number of global communications that take very less time compared to that of the computations which is evident from the parallel scaling results in Section 5.4.

4.6.3 Memory requirements

In this matrix-free implementation of the method, there is no need for storage of any dense or sparse matrices. We store the arrays for $b(\mathbf{x})$, $\phi(\mathbf{x})$, $V_{\text{eff}}(\mathbf{x})$, $\rho(\mathbf{x})$, atom positions within some distance of Ω_P and their non-local projectors etc., distributed across all the processors. Every processor stores all the atom positions in Ω and their forces. Also, every processor P, stores the Chebyshev coefficients (c_p^j), components (ρ_p^j) and scaling factors for all $p \in K_{\Omega_P}$ and for $j \in \{0 \text{ to } n_{pl}\}$. So the storage in each processor scales as $\mathcal{O}((c+n_{pl})N_{proc})$, where N_{proc} is the number of grid points in the processor domain Ω_P and c is some constant to account for all the other arrays distributed across processors. Hence, memory requirement also scales linearly with system size. For simulations involving high prefactor ($c << n_{pl}$), one can compute Fermi energy (Eqn. 120) without the need to even store the Chebyshev coefficients and components. In such a case, the trade-off would be to perform more computations to evaluate them in every iteration of Fermi energy calculation.

CHAPTER V

RESULTS AND EXAMPLES

In this section, we verify the convergence of the Clenshaw-Curtis Spectral Quadrature (SQ) method for canonical insulating and metallic systems. Specifically, we demonstrate convergence with respect to quadrature order and truncation radius to the diagonalization result. We also show convergence with respect to mesh size to established $\mathcal{O}(N^3)$ scaling planewave results. To allow rigorous assessment of convergence, we first consider a model one-dimensional problem in Section 5.1. We then consider full, three-dimensional Kohn-Sham calculations in Section 5.2. In Section 5.3, we study the efficiency of SQ for performing high temperature calculations and discuss the parallel scaling studies in Section 5.4. We also compare computational timings with a diagonalization based code and show some preliminary results for ab-initio molecular dynamics simulation.

5.1 Validation through a model problem

We begin with a one-dimensional model problem wherein atoms interact with the effective potential [31]

$$V_{\mathit{eff}}(x, \mathbf{R}) = -\sum_J \frac{\alpha}{\sqrt{2\pi\beta^2}} \exp\left(-\frac{(x - R_J)^2}{2\beta^2}\right). \tag{131}$$

We choose the parameters $(\alpha, \beta) = (10, 0.6)$, and consider an infinite chain of atoms with unit lattice constant. In addition, we employ a twelfth order finite-difference discretization with a mesh-size of $h = 0.1$, and a smearing of $\sigma = 1$. In Fig. 15, we show the convergence of the Γ-point energy (obtained via diagonalization) to the infinite-cell result as the number of unit cells (containing one atom each) in Ω is

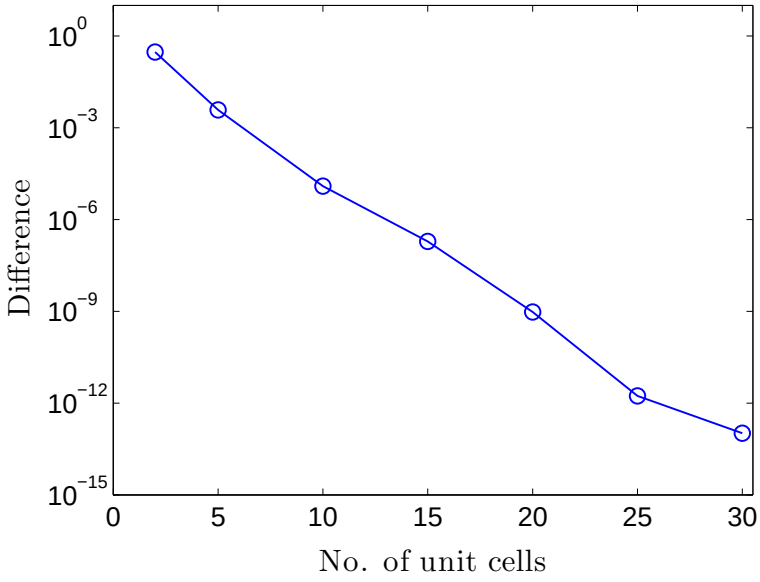
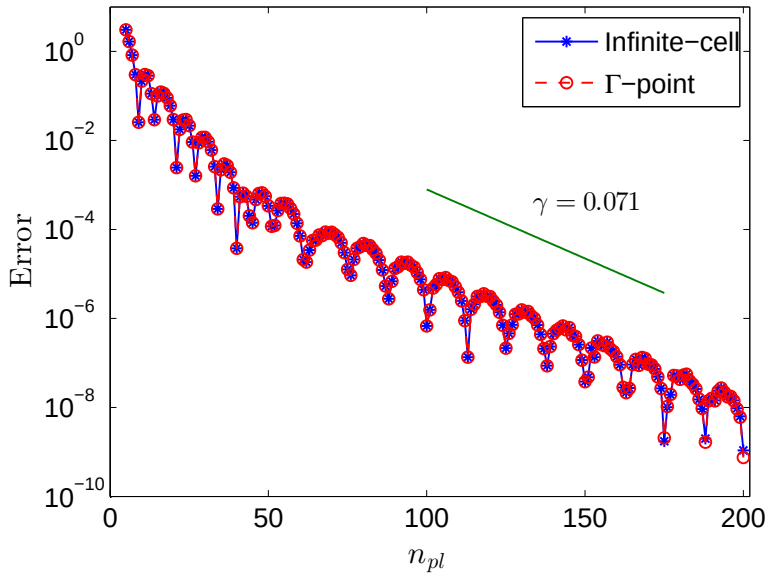


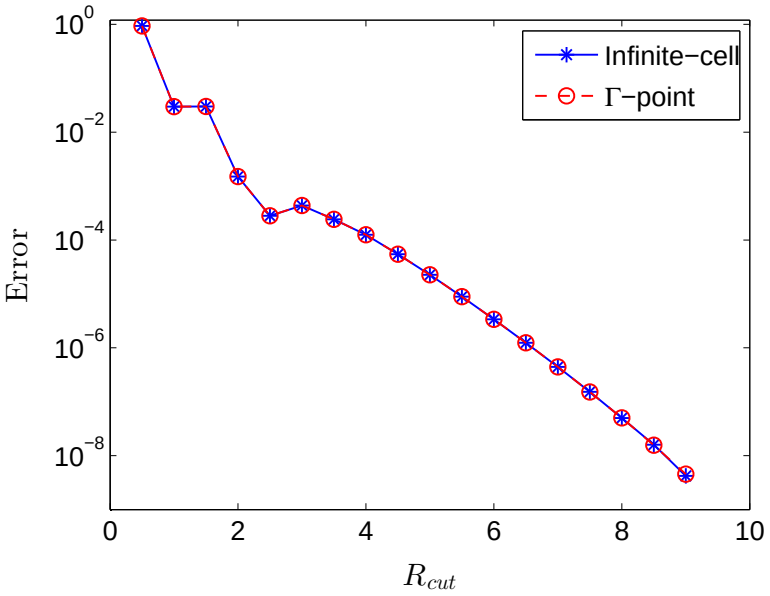
Figure 15: Convergence of the Γ-point energy to the infinite-cell limit for a 1D model problem. The energy corresponding to a 40 unit-cell system has been used as reference.

increased. We see that the energy has converged to within $\sim 10^{-12}$ of the infinite-unit-cell limit for Ω containing 25 unit cells. In order to validate the SQ method, we therefore choose 2-atom and 25-atom cells for the infinite-cell and Γ-point calculations, respectively. The converged energies for each case should then differ by no more than $\sim 10^{-12}$.

The two parameters introduced by the Clenshaw-Curtis SQ method are the order of the quadrature n_{pl} and the truncation radius R_{cut}. In Fig. 16, we show the convergence with respect to these two parameters for both Γ-point and infinite-cell calculations. Specifically, we plot the error in energy—defined with respect to the result obtained by diagonalization—versus n_{pl} and R_{cut} in Figs. 16a and 16b, respectively. Since the cell Ω in the Γ-point calculations has been chosen sufficiently large to reduce finite-size effects to $\sim 10^{-12}$, the Γ-point and infinite-cell results should be essentially indistinguishable, as we see in Fig. 16. We observe that SQ demonstrates



(a) Convergence with n_{pl}



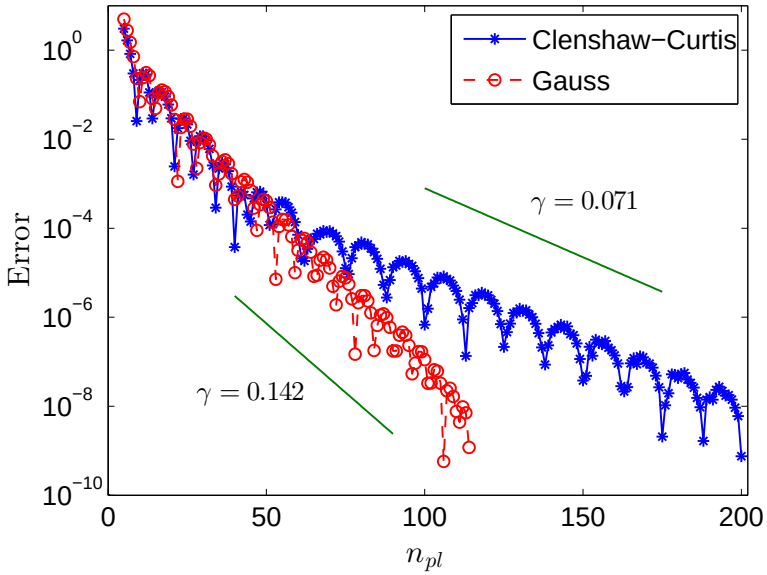
(b) Convergence with R_{cut}

Figure 16: Convergence of energy with respect to quadrature order n_{pl} and truncation radius R_{cut} to the diagonalization result for a 1D model problem. 2-atom and 25-atom unit cells have been employed for the infinite-cell and Γ-point calculations, respectively. The theoretically predicted convergence rate [103] is denoted by γ.

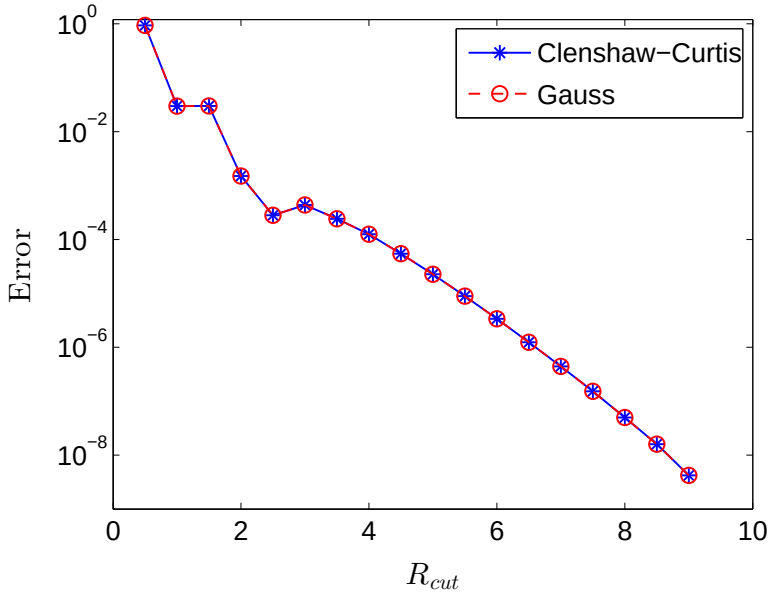
exponential convergence with respect to n_{pl}, with the obtained rate in excellent agreement with theoretical predictions [103]. Moreover, there is exponential convergence with respect to R_{cut}, again in agreement with theory [10].

Apart from the Clenshaw-Curtis SQ method developed in the present work, the other notable variant of SQ is based on Gauss quadrature [103]. For the 1D model problem, in Figs. 17a and 17b, we compare the convergence of the two approaches with respect to n_{pl} and R_{cut}, respectively. These results correspond to Γ-point (convergence with n_{pl}) and infinite-cell (convergence with R_{cut}) calculations for 25-atom and 2-atom unit cells, respectively. We observe that the asymptotic convergence rate of Gauss SQ is twice that of Clenshaw-Curtis SQ, and both are in good agreement with theoretical predictions [103]. Notably, the initial convergence rate of Clenshaw-Curtis SQ is twice its asymptotic value, a phenomenon which has been discussed in detail elsewhere [110]. The convergence of both methods with respect to R_{cut} is identical, as dictated by the decay of the density matrix.

In Clenshaw-Curtis SQ, the key computations occurring for every finite-difference node are the n_{pl} local matrix-vector multiplications, and evaluation of the n_{pl} Chebyshev expansion coefficients. In practical DFT calculations, the relatively large nodal Hamiltonians result in the matrix-vector products being the dominant computational cost. Similarly, the dominant cost in Gauss SQ corresponds to the n_{pl} local matrix-vector multiplications during the Lanczos iteration, with the tridiagonal matrix orthogonalization constituting a relatively small fraction of the total expense. It is evident from Fig. 17a that the quadrature order required by Gauss SQ is typically smaller than for Clenshaw-Curtis SQ. However, since both approaches display similar convergence behavior in the initial stages, the quadrature orders required for achieving chemical accuracies do not differ significantly. In this work, we have preferred the Clenshaw-Curtis variant of SQ since the nonlocal component of the atomic forces can be evaluated straightforwardly and efficiently (sub-section 4.5.2), whereas the path to



(a) Convergence with n_{pl}



(b) Convergence with R_{cut}

Figure 17: Comparison of the convergence of the Clenshaw-Curtis and Gauss SQ methods to the diagonalization result for a 1D model problem. 2-atom and 25-atom unit cells have been employed for the infinite-cell (convergence with R_{cut}) and Γ-point (convergence with n_{pl}) calculations, respectively. The theoretically predicted convergence rate [103] is denoted by γ.

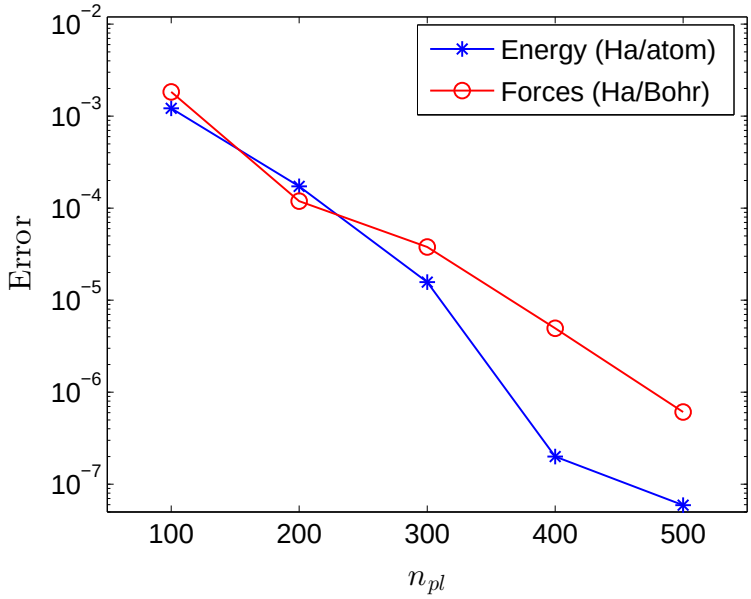
such is unclear for Gauss SQ. In addition, the evaluation of the Chebyshev coefficients in Clenshaw-Curtis SQ scales as $\mathcal{O}(n_{pl} \log n_{pl})$ [16], whereas orthogonalization of the tridiagonal matrix in Gauss SQ scales as $\mathcal{O}(n_{pl}^2)$ [26].

5.2 Accuracy and convergence

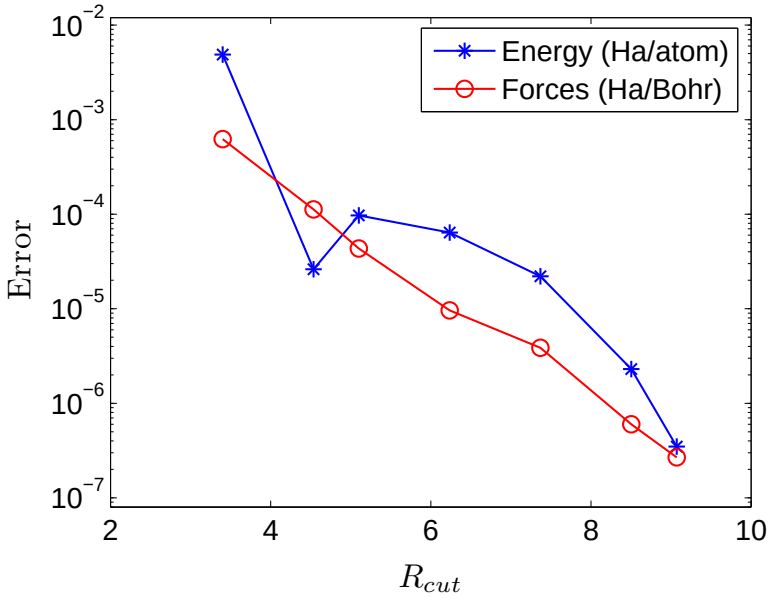
We consider two systems for the Kohn-Sham calculations: (i) lithium hydride, a prototypical insulator, and (ii) aluminum, a prototypical metal. In all simulations, we use norm-conserving Troullier-Martins pseudopotentials [111], and the Perdew-Wang parametrization [81] of the correlation energy calculated by Ceperley-Alder [22]. Additionally, we utilize a smearing of 0.5 eV and 4 eV for the Γ-point and infinite-cell calculations, respectively. Finally, we employ a twelfth order accurate finite-difference discretization, unless otherwise specified.

5.2.1 Lithium hydride

We consider an 8-atom unit cell of lithium hydride at the equilibrium lattice constant of 7.37 Bohr. We displace the lithium atom at the corner of the unit cell, which corresponds to the origin of the chosen coordinate system, to [0.57 0.43 0.37] Bohr. For lithium's nonlocal pseudopotential, we designate the s channel as local, and utilize cutoff radii of 2.43 Bohr for both the s and p channels. For hydrogen's pseudopotential, we use a cutoff radius of 1.4 Bohr. We perform both Γ-point and infinite-cell calculations with $h = 0.57$ Bohr, and present the results so obtained in Fig. 18. Specifically, we plot the convergence of SQ energy and forces with respect to quadrature order n_{pl} and truncation radius R_{cut} for Γ-point and infinite-cell calculations in Figs. 18a and 18b, respectively. The errors are defined relative to the results obtained by diagonalization. As in the case of the one-dimensional model problem, we obtain exponential convergence with respect to both parameters. These results indicate that the theoretical predictions made in the linear setting [103, 10] are also applicable to the nonlinear Kohn-Sham problem. We note that neither energies nor



(a) Γ-point calculation



(b) Infinite-cell calculation

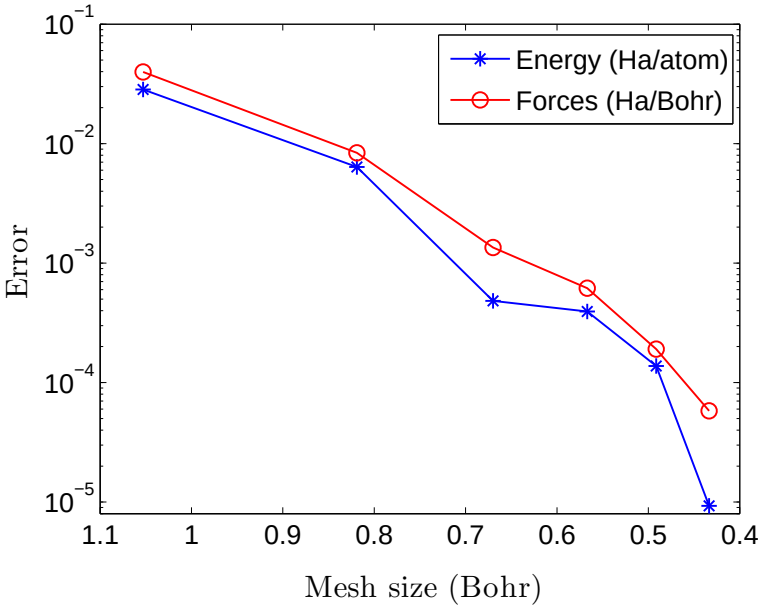
Figure 18: Convergence of energy and forces with respect to quadrature order n_{pl} and truncation radius R_{cut} to the diagonalization result for lithium hydride. The error in energy denotes the magnitude of the difference, and error in forces represents the maximum difference in any component.

forces are variational with respect to R_{cut}, and so errors can in general be positive, negative, or zero. This can be seen, for example, in the non-monotonic convergence of the energy error in Fig. 18b.

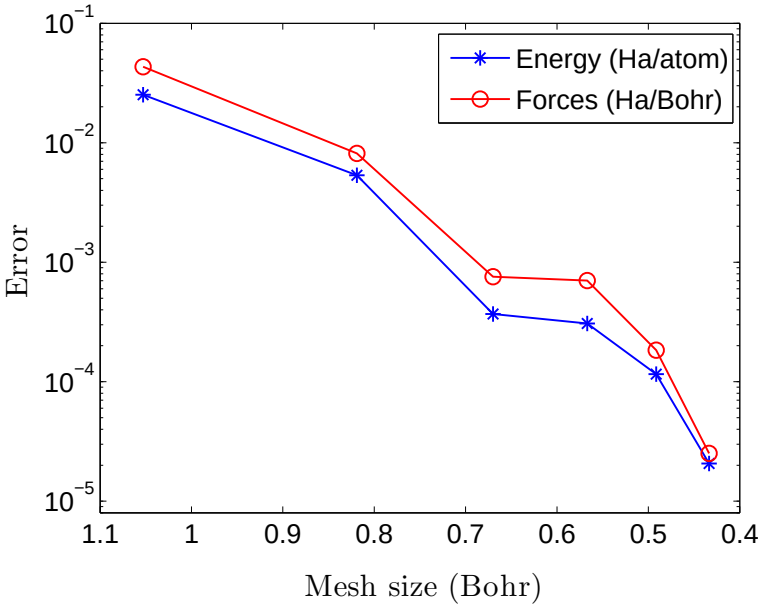
Next, we verify the accuracy of SQ by comparing the calculated energies and forces to those computed by the ABINIT planewave code [40]. In ABINIT, we employ a planewave cutoff of 50 Ha, and a $6 \times 6 \times 6$ Monkhorst-Pack grid for Brillouin zone integration. These parameters result in energies and forces converged to within 10^{-6} Ha/atom and 10^{-6} Ha/Bohr, respectively. In SQ, wherever it is not possible to utilize twelfth order finite-differences, we employ the largest order feasible. Additionally, we utilize $\{n_{pl}, R_{cut}\} = \{550, 3.69\,\text{Bohr}\}$ and $\{n_{pl}, R_{cut}\} = \{160, 8.50\,\text{Bohr}\}$ for the Γ-point and infinite-cell calculations, respectively. These values are sufficient to put the associated errors well below the mesh errors of interest. We note that a larger value of n_{pl} is required for the Γ-point calculation because of the lower value of smearing/temperature. As shown in Fig. 19, both energies and forces in SQ converge rapidly and systematically, with chemical accuracy easily obtained. Notably, we see that energies and forces converge at comparable rates, without need of additional measures such as double-grid [73] or high-order integration [11] techniques. Hence, accurate forces are easily obtained, as required for structural relaxation and molecular dynamics simulations.

5.2.2 Aluminum

We now consider a 4-atom face-centered cubic (FCC) unit cell of aluminum at the equilibrium lattice constant of 7.78 Bohr. We move the atom located at [3.89 3.89 0.00] Bohr to [3.74 3.49 0.37] Bohr, with the corner atom again coinciding with the origin. We utilize a nonlocal pseudopotential having cutoff radius of 2.58 Bohr for both the s and p channels, with the p channel chosen as local. In Fig. 20, we show

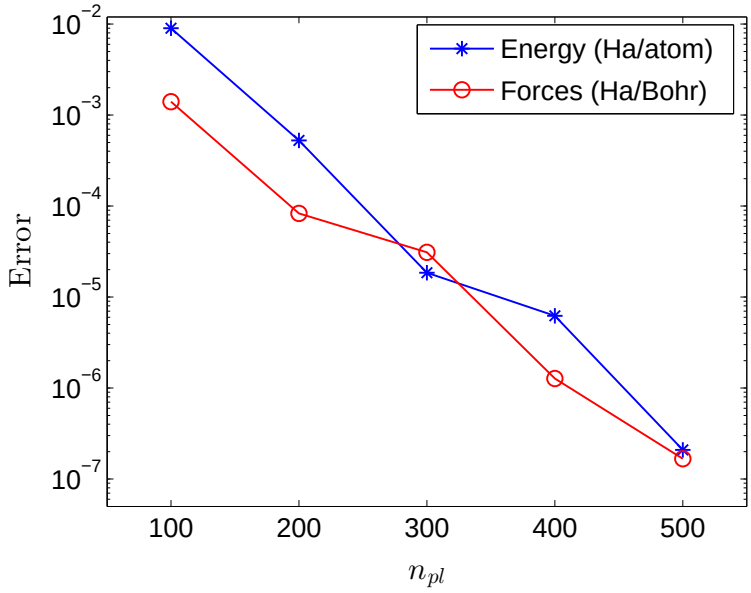


(a) Γ-point calculation

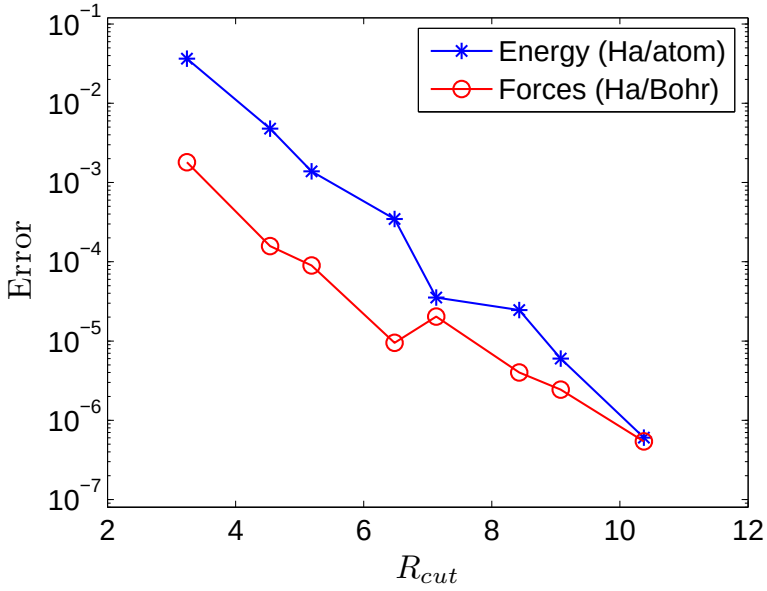


(b) Infinite-cell calculation

Figure 19: Convergence of energy and forces with respect to mesh size to reference planewave result for lithium hydride. The error in energy denotes the magnitude of the difference, and error in forces represents the maximum difference in any component.



(a) Γ-point calculation



(b) Infinite-cell calculation

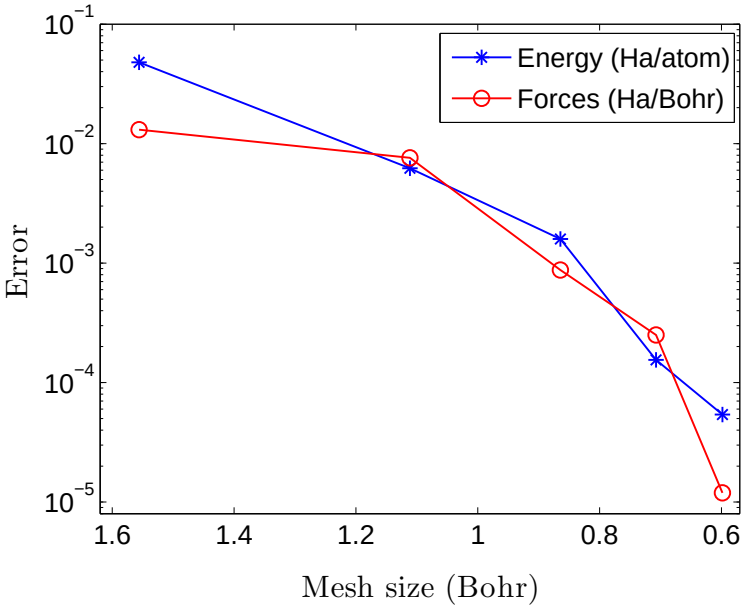
Figure 20: Convergence of energy and forces with respect to quadrature order n_{pl} and truncation radius R_{cut} to the diagonalization result for aluminum. The error in energy denotes the magnitude of the difference, and error in forces represents the maximum difference in any component.

the convergence of energy and forces with respect to quadrature order n_{pl} for Γ-point calculations, and truncation radius R_{cut} for infinite-cell calculations. We choose $h = 0.65$ Bohr for the calculations and define all errors with respect to diagonalization. For this prototypical metallic system, we obtain exponential convergence with respect to both parameters, just as for the insulating lithium hydride system.

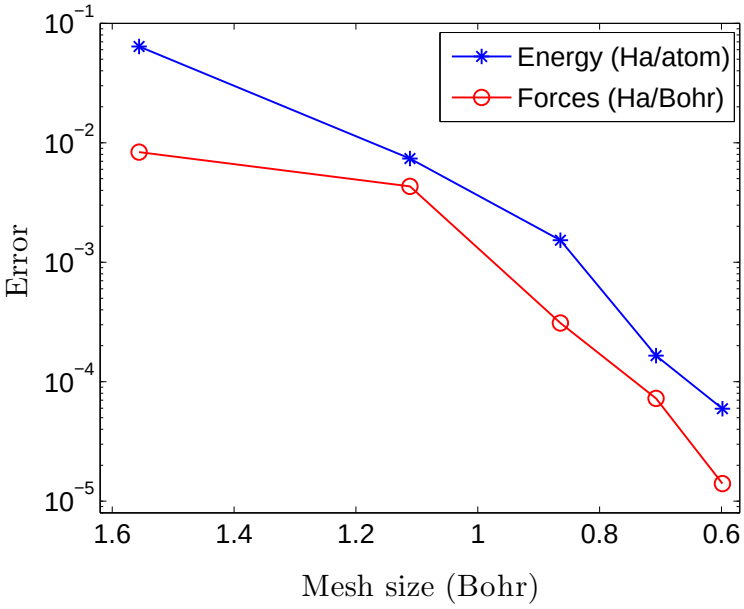
Finally, we compare the results obtained by SQ and ABINIT for the aforedescribed aluminum system. In ABINIT, we utilize a planewave cutoff of 30 Ha, and a $6 \times 6 \times 6$ Monkhorst-Pack grid for Brillouin zone integration. The energies and forces so computed are converged to within 10^{-6} Ha/atom and 10^{-6} Ha/Bohr, respectively. In SQ, whenever n_d is too small to utilize twelfth order finite-differences, we again employ the largest order possible. Additionally, we utilize $\{n_{pl}, R_{cut}\} = \{600, 3.89\,\text{Bohr}\}$ and $\{n_{pl}, R_{cut}\} = \{120, 10.00\,\text{Bohr}\}$ for the Γ-point and infinite-cell calculations, respectively. It is clear from the results presented in Fig. 21 that similar to the case of lithium hydride, both energies and forces converge rapidly, and at similar rates. Overall, we see that SQ is able to obtain chemical accuracy in both energies and forces, straightforwardly and systematically, in both insulating and metallic systems.

5.3 Efficiency at higher temperatures

The computational cost of conventional diagonalization-based DFT calculations grows rapidly with increasing temperature due to a larger number of previously unoccupied states becoming partially occupied. Since most diagonalization algorithms scale quadratically with the number of states to be computed, high temperature calculations can quickly become intractable [116]. However, in the Clenshaw-Curtis SQ method, the overall cost *decreases* with increasing temperature. This is due to the enhanced decay of the density matrix, which translates to smaller values of the truncation radius R_{cut}. Furthermore, a lower-order quadrature rule suffices due to the increased smoothness of the Fermi-Dirac function, resulting in additional savings. In



(a) Γ-point calculation



(b) Infinite-cell calculation

Figure 21: Convergence of energy and forces with respect to mesh size to reference planewave result for aluminum. The error in energy denotes the magnitude of the difference, and error in forces represents the maximum difference in any component.

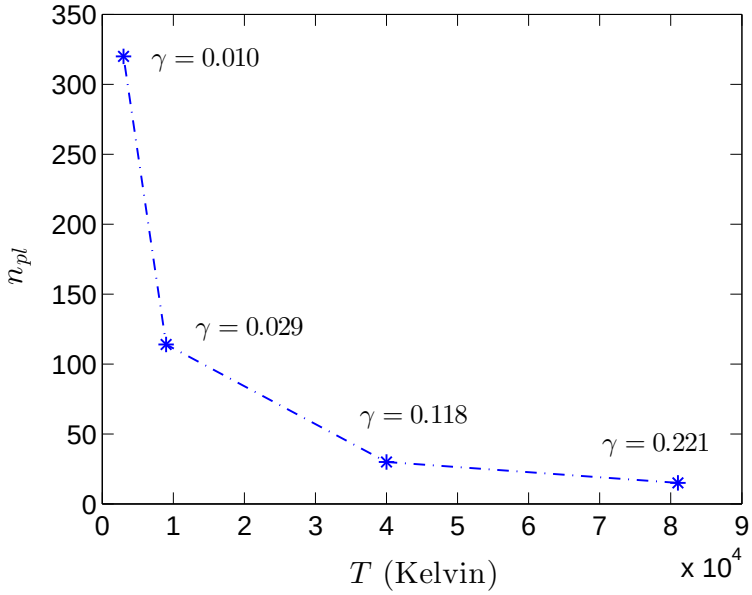


Figure 22: Dependence of quadrature order n_{pl} on temperature T for errors of ∼0.0001 Ha/atom in energy and ∼0.0001 Ha/Bohr in forces. The theoretically predicted convergence rate [103] is denoted by γ.

order to quantify this reduction in quadrature order, we consider the Γ-point calculation for the aluminum system described in sub-section 5.2.2 at different temperatures, and determine the order of quadrature required to attain convergence of $\sim$ 0.0001 Ha/atom in the energy and $\sim$ 0.0001 Ha/Bohr in atomic forces. We present the results so obtained in Fig. 22, along with theoretically predicted convergence rates γ [103]. We observe that there is indeed a rapid decrease in the order of quadrature, and thus computational cost, required to obtain the specified accuracies as temperature increases. Moreover, the quadrature order required is consistent with the predicted convergence rate γ [103], which increases with temperature.

Since the required order of quadrature in SQ varies inversely with the temperature for a given spectral width [103], it can be inferred from Fig. 22 that quadrature orders of a few thousand are required for calculations near room temperature. Therefore, the SQ approach possesses a relatively large prefactor for such temperatures and/or Hamiltonians with large spectral width. However, the particular suitability of SQ for

large-scale parallel computation, as discussed below, stands to mitigate this. Therefore, it may be expected that calculations at room temperature will be feasible using Clenshaw-Curtis SQ when sufficient processors are available. One helpful degree of freedom to exploit in practice is the use of larger occupation smearing than $\sigma = 0.001$ Ha ($T \sim 315$ K) as an approximation. The typical practice in metallic calculations of employing a smearing of $\sigma \sim 0.01$ Ha ($T \sim 3150$ K), for example, could yield substantial efficiency gains, while retaining energies and forces to high accuracy relative to $\sigma = 0.001$ Ha values.

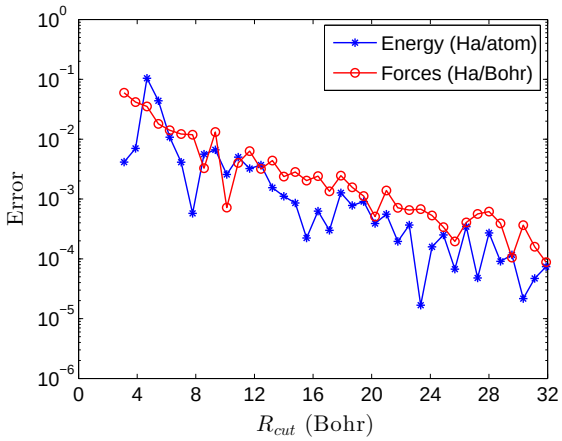
In order to study the dependence of the truncation radius (R_{cut}) with the temperature, we consider the infinite-cell calculation of the aluminum system. Since, the localization depends on the decay of density matrix, we expect that the truncation radius required for a particular level of accuracy to reduce with increasing temperature. In Fig. 23, we plot the convergence of energy and forces with respect to R_{cut} at various temperatures (smearings). The error is computed with respect to a converged value with respect to R_{cut} at each of the temperatures. The R_{cut} required for an accuracy of 0.0001 Ha/atom or Ha/Bohr is about 32 Bohr at $T = 2368$ K, 16 Bohr at $T = 12630$ K and 10 Bohr at $T = 31577$ K, demonstrating a decreasing trend with temperature.

5.4 Parallel scaling studies

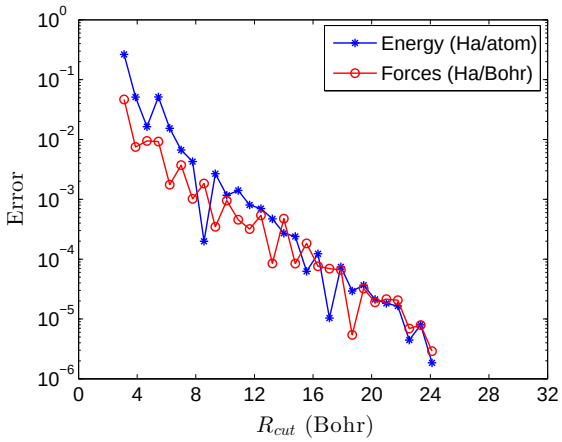
The parallel simulations have been performed on the Lawrence Livermore National Laboratory's computational resources *vulcan*[1] and *sierra*[2].

Vulcan[3] is an IBM Blue Gene/Q system with PPC A2 CPUs, BlueGene torus high-speed interconnect, 1.6 GHz CPU speed, 42.6 GB/s peak CPU memory bandwidth and 5033 TFLOP/s peak performance. It has 24576 compute nodes with 16 cores

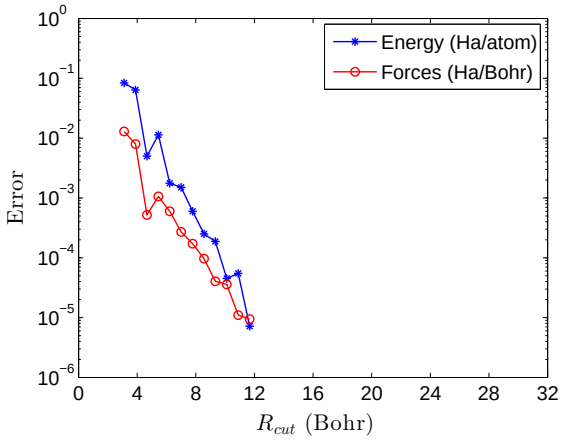
[1] http://computation.llnl.gov/computers/vulcan
[2] https://computing.llnl.gov/?set=resources&page=OCF_resources#sierra
[3] https://computing.llnl.gov/?set=resources&page=OCF_resources#vulcan



(a) $T = 2368$ K



(b) $T = 12630$ K



(c) $T = 31577$ K

Figure 23: Convergence of energy and forces with respect to localization radius R_{cut} at various temperatures for Aluminum.

each, with a total of 393216 compute cores.

Sierra has Intel Xeon EP X5660 CPUs, InfiniBand QDR (QLogic) high-speed interconnect, 2.8 GHz CPU speed, 32 GB/s peak CPU memory bandwidth and 261.3 TFLOP/s peak performance. It has nodes with 12 cores each and a total of 23328 cores.

Conventional $\mathcal{O}(N^3)$ scaling diagonalization-based electronic structure methods involve the computation of large numbers of orthonormal eigenvectors, which requires communication between each processor holding all or a part of an eigenvector and every other processor holding all or a part of other eigenvectors. This extensive communication severely limits parallel scalability. $\mathcal{O}(N)$ electronic structure methods compute required densities, energies, and forces, proceeding directly from Hamiltonian to required outputs without diagonalization. The key computational workload in standard $\mathcal{O}(N)$ electronic structure methods such as Fermi operator expansion (FOE) [38, 39] instead comes in the form of repeated large sparse matrix-matrix multiplies involving the Hamiltonian. However, with each multiply, the sparsity pattern changes, necessitating truncation to retain only desired nonzeros. Moreover, nonlocal communications and indirect addressing are required to map nonzeros of corresponding rows and columns to one another, making such operations difficult to parallelize at large scale.

Being integral based, however, the SQ electronic structure method admits a natural decomposition in real-space, eliminating the need for large sparse matrix-matrix operations entirely. To exploit this, we have represented the electronic densities and potentials on a uniform finite-difference grid in real-space. Moreover, we have exploited the locality of the density matrix to replace large global sparse matrix-matrix operations by small local sparse matrix-vector operations (computed in a matrix free way), with all calculations strictly confined to localization regions around each grid

point. In so doing, the key computational operations are reduced to local finite-difference stencil operations, global vector sums, and dot products. Storage and memory access can be minimized by forming local matrices on the fly, simultaneously at each grid point, and can be reduced further still by computing only the action of matrices on required vectors, rather than computing the matrices themselves. These latter possibilities arise naturally in the SQ formulation, allowing flops to be traded for decreased storage, access, and/or communication, as best suits the architecture at hand. By reducing all key computational kernels to local stencil and vector operations, the SQ method is well suited to large-scale parallel implementation.

In this Section, we look at the strong and weak scaling of the FCC Aluminum system with 4 eV smearing. The equilibrium lattice constant is 7.78 Bohr. The four atoms in the unit cell are located at [0.00 0.00 0.00] Bohr, [0.00 3.89 3.89] Bohr, [3.89 0.00 3.89] Bohr and [3.74 3.49 0.37] Bohr. In all the scaling simulations, we discretize the domain with a mesh size of 0.778 Bohr and choose the SQ paramters $n_{pl} = 28$ and $R_{cut} = 6$ Bohr, which are all sufficient to give an accuracy of below 0.001 in energy (Ha/atom) and forces (Ha/Bohr). We only consider the infinite-cell calculations in all the subsequent simulations. The parallel code has been written using MPI and C.

5.4.1 Strong scaling

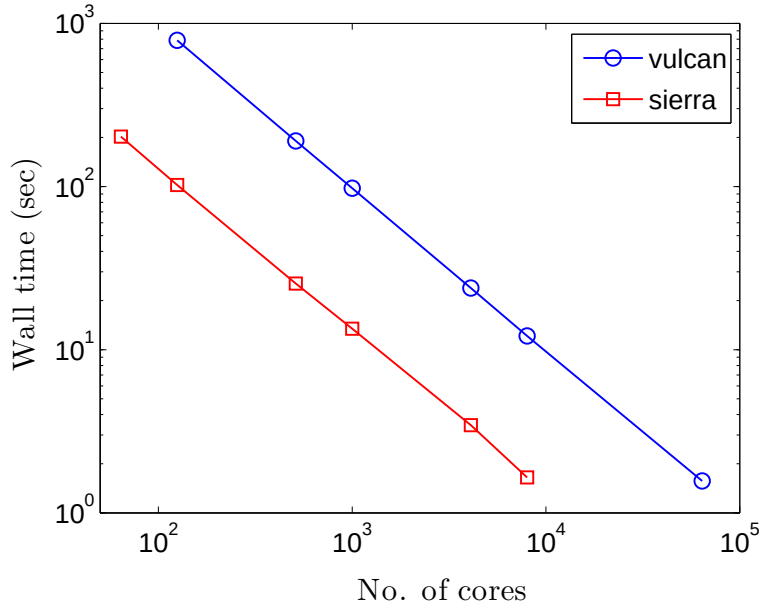
In strong scaling, we are interested in looking at the efficiency of speeding up the calculations (i.e. reducing the wall time) as the number of processors allocated for the problem are increased. Strong scaling determines how fast a given problem can be solved using the maximum number of resources. With more number of processors, the number of grid points per processor reduces and hence the amount of computation to be carried out by any processor also reduces. If a calculation involves minimal communication time compared to that of computations, then the efficieny of reducing the wall time proportional to the increase in number of processors is very high. This

efficiency would be affected by both local and global communications.

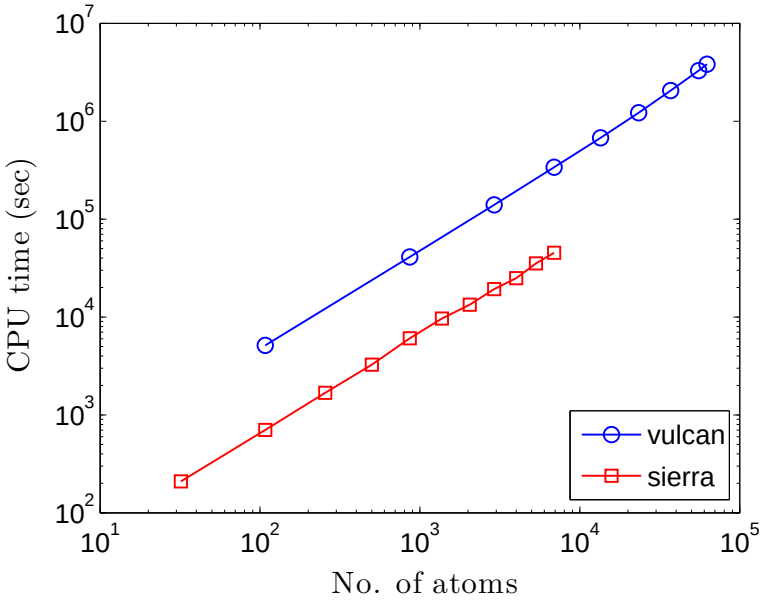
For the strong scaling study, we choose a 2048 atoms system obtained by replicating the unit cell 8 times in each direction. In typical calculations, solving the SCF is the dominant cost and hence scaling of a single SCF iteration would determine the scaling of the overall calculation. In Fig. 24a, we present the strong scaling of the method by plotting the wall time taken by a single SCF iteration. The system under consideration has $80^3 = 512000$ grid points. As we go from 64 processors to 8000 processors, the number of grid points per processor changes from 20^3 to 4^3 respectively. On the vulcan cluster, we go until there are 2^3 grid points per processor. The calculation shows a very good strong scaling with about 98% efficiency (on both vulcan and sierra) at the lowest point in the plot. This shows the near perfect strong scaling behavior of the method and can be attributed to the local nature of the spectral quadrature method which takes most of the time in a single SCF iteration. If we include the total SCF and force calculation time, the efficiency is about 93% on sierra and 60% on vulcan at the lowest points in the plot. At 64000 processors on vulcan, the reduced efficiency is mainly due to the loss of strong scaling in force calculation as the number of grid points in each direction in the processor has become much smaller than the finite difference stencil width (see Table 9). Notice that in Table 9, the time taken by SCF decreases by about 7 times from 8000 to 64000 processors (8 fold increase) but the time taken by forces decreases by only about 3 times.

Table 9: Wall times (in seconds) for the strong scaling study of 2048 atoms Aluminum system using SQ method on vulcan cluster. The timings for SCF, forces and the timings for a single SCF iteration are presented in the above table.

n_{proc}	SCF	Forces	One SCF iteration
125	4734	1228	788
512	1143	298	190
1000	597	161	98
4096	143	42	24
8000	73	25	12
64000	10.7	8.7	1.6



(a) Strong scaling



(b) Weak scaling

Figure 24: Parallel scaling study of the Spectral Quadrature $\mathcal{O}(N)$ DFT method. The computational times in the plots indicate the time taken by a single SCF iteration. (a) The efficiency at the lowest point in strong scaling is about 98%. (b) The slopes of linear fits for vulcan and sierra in weak scaling are 1.037 and 1.000 respectively.

5.4.2 Weak scaling

In weak scaling, we are interested in determining the ability of the method to solve larger problems with the same amount of computational effort per the allocated resource. Given a problem that can be solved on a particular number of processors within a specified time, weak scaling determines how big a problem can be solved in almost the same time utilizing proportionally larger number of resources. Since we use a scalable routine (see sub-section 4.6.2) for local communications, the weak scaling would only be hindered by global communications that increase proportionally with number of processors. So, if the computations are much more significant than global communications (which is typically the case, unless we are at the limits of strong scaling for a large problem), we expect that the weak scaling has the same behavior as the overall computational scaling of the method (i.e. $\mathcal{O}(N)$).

For the weak scaling study, we choose 8 processors to solve the 4 atom unit cell where 5^3 grid points are allocated to each processor. Larger systems are obtained by replicating this unit cell and the associated number of processors in order to maintain the number of grid points per processor at a constant value of 5^3. This choice of grid points per processor corresponds to a strong scaling efficiency of 98% and 91% on vulcan and sierra respectively for the total SCF and force calculation. On the sierra cluster, we start from 32 atoms on 64 processors and go until 6912 atoms on 13824 processors. On the vulcan cluster, we start from 108 atoms on 216 processors and go until 62500 atoms on 125000 processors. Fig. 24b shows the weak scaling of the method with CPU time plotted as the size of the system increases. CPU time is the total time taken by all the processors which is given by the product of wall time and the number of processors. The scaling laws (based on the time for single SCF iteration) on vulcan and sierra are computed to be $\mathcal{O}(N^{1.037})$ and $\mathcal{O}(N^{1.000})$ respectively. If the scaling laws are calculated based on the total time for SCF and forces then we get, $\mathcal{O}(N^{1.035})$ and $\mathcal{O}(N^{1.014})$ on vulcan and sierra respectively. As

expected, this is in agreement with the $\mathcal{O}(N)$ scaling of the method.

5.5 Comparison with diagonalization

In this section, we compare the performance of SQ method with respect to a diagonalization based code ABINIT [40] at high temperature for the Aluminum system used in previous section. We employ a smearing of 4 eV which corresponds to an electronic temperature of 46418.02 K. In Table 10, we compare the computational time taken by the SQ method and ABINIT for two different system sizes. The convergence parameters and mesh sizes (for both SQ and ABINIT) chosen are just sufficient to give the chemical accuracy of 0.001 Ha/atom in energy and 0.001 Ha/Bohr in forces. Specifically, we choose a mesh size of 0.778 Bohr in SQ and a planewave cut-off of 8 Ha in ABINIT. Large size systems have been obtained by replicating the 4-atom unit cell in the three directions. The 108 atoms system needs about 1400 states and the 500 atom system needs about 5000 states in ABINIT simulations. The number of processors for both SQ and ABINIT have been chosen to give minimum wall time, restricting the computational resources to 1024 processors. SQ method implemented in parallel shows about one and two orders of magnitude speedup over ABINIT for the 108 and 500 atoms systems respectively. It can be noted that the speedup is higher for the larger system. We also note that the SQ wall times will be even lower if more processors are used. The simulations are performed on the computing nodes with the following configuration: Altus 1804i Server - 4P Interlagos Node, Quad AMD Opteron 6276, 16C, 2.3 GHz, 128GB, DDR3-1333 ECC, 80GB SSD, MLC, 2.5" HCA, Mellanox ConnectX 2, 1-port QSFP, QDR, memfree, CentOS, Version 5, and connected through InfiniBand cable.

Table 10: Comparison of minimum wall time (in seconds) for SQ and ABINIT. The numbers in the parentheses indicate the number of processors for which we obtained minimum wall time for the respective systems. N is the number of atoms in the system.

N	SQ	ABINIT
108	28 (1000)	401 (360)
500	51 (1000)	9736 (600)

5.6 Molecular dynamics

In Molecular Dynamics (MD), one is interested in propagating the system in discrete time steps and finding out any properties of interest through time averaging. In Ab-Initio Molecular Dynamics (AIMD) or Quantum Molecular Dynamics (QMD), the interactions between the atoms are computed from first principles. We employ Born-Oppenheimer Molecular Dynamics, where the nuclear motion is decoupled from the electronic structure problem [67]. We solve for the electron density and potential using the SQDFT method for a fixed position of the nuclei. The nuclear positions are then updated based on the forces calculated from DFT. Below, we discuss some preliminary results for an AIMD simulation.

We perform NVE (also called microcanonical ensemble) simulation where the total energy of the system remains constant with time. The atomic positions are updated using the Leapfrog method which gives a second order approximation in time discretization [94]. The atoms that move outside the simulation domain Ω are periodically mapped back inside. All the atoms are assigned an initial velocity with uniform magnitude (and random direction) based on the initial temperature of 46418.02 Kelvin (corresponding to a smearing of 4 eV). The initial accelerations are set to zero. We choose an MD time step of 0.1 femto seconds (fs) and average over a maximum of

500 MD steps. The total energy (TE) of the system at any time step is given by the sum of free energy (FE) and kinetic energy (KE). The time step has been chosen such that it keeps the discretization error within some tolerance that conserves the total energy. Since the total energy is to be conserved, the change in FE of the system as the atoms move around is converted to KE (related to the velocities of the atoms) which changes the instantaneous temperature of the system. We use a charge extrapolation algorithm to provide a good guess for the electron density in each MD step for faster convergence of the SCF [2].

In Table 11, we present the average and standard deviations of the energies at a few intermediate MD steps for the 108 atom Aluminum system discretized with 0.778 Bohr mesh. This mesh gives an error of less than 0.001 in energy (Ha/atom) and forces (Ha/Bohr). Moreover, the forces are consistent with energy to about 0.001 Ha/atom, which is required for the conservation of energy. The average of the sum of forces on an atom is within the 0.0001 Ha/Bohr and this conserves the linear momentum of the system. Consequently, the average of sum of velocities of an atom at any step is also below 0.0001 Bohr/fs. The averages are computed from across all the atoms in Ω. We can observe from Table 11 that the KE of the system has reduced gradually while the FE has increased. This has caused a reduction in the temperature of the system. The SQ parameters R_{cut} and n_{pl} are chosen such that they give sufficient accuracy even when the temperature of the system has dropped.

Table 11: The average and standard deviation of the total energy (TE), free energy (FE) and kinetic energy (KE) at different steps of a 108 Al atom NVE MD simulation. The energy units are in Ha/atom.

	Total Energy		Free Energy		Kinetic Energy	
Time step	Average	Std. dev.	Average	Std. dev.	Average	Std. dev.
100	−2.2384	0.0007	−2.4389	0.0160	0.2005	0.0157
200	−2.2383	0.0006	−2.4017	0.0403	0.1634	0.0401
300	−2.2383	0.0005	−2.3900	0.0368	0.1517	0.0367
400	−2.2383	0.0004	−2.3858	0.0327	0.1475	0.0327
500	−2.2383	0.0004	−2.3840	0.0295	0.1457	0.0294

CHAPTER VI

CONCLUSION

This work is motivated by a need to simulate large systems of atoms based on first principles to model materials at nanoscale. Current codes that can simulate metals scale cubically with number of atoms in the system and hence cannot tackle very large systems with many hundreds or thousands of atoms. In this work, we used the density matrix formulation of Density Functional Theory (DFT) to develop a linear scaling method based on Spectral Quadrature technique. We have presented the Clenshaw-Curtis Spectral Quadrature (SQ) method for performing real-space $\mathcal{O}(N)$ Density Functional Theory (DFT) calculations. In this approach, all quantities of interest are expressed as bilinear forms, or sums over bilinear forms, which are then approximated by spatially localized Clenshaw-Curtis quadrature rules. In conjunction with the local reformulation of the electrostatics, the proposed approach enables the $\mathcal{O}(N)$ evaluation of the electronic density, energy, and atomic forces. In addition, the method permits infinite-cell calculations without recourse to Brillouin zone integration or large supercells.

We also discuss acceleration techniques for fixed-point iterations, which are a key component of DFT simulations. We propose new techniques to accelerate the SCF which is a non-linear fixed point iteration in DFT. We also apply those techniques to linear fixed point iteration obtained from Jacobi iteration to develop efficient solvers for linear systems of equations. For all the proposed methods we discuss and demonstrate their performance through examples. In the DFT simulations, we chose the Alternating Anderson technique as the Poisson solver and for acceleration of SCF due its amenability to massive parallelization.

We discuss the parallel numerical implementation of the SQ method where we have employed a high-order finite difference representation in order to exploit the locality of electronic interactions in real-space, enable systematic convergence, and facilitate large-scale parallel implementation. In this representation, we have developed expressions for the electronic density, total energy, and atomic forces which can be evaluated in $\mathcal{O}(N)$ operations. The SQ energies and forces were shown to converge systematically with respect to quadrature order and truncation radius to the exact diagonalization result, for 1D model as well as full 3D Kohn-Sham calculations of insulating and metallic systems. Moreover, convergence to established $\mathcal{O}(N^3)$ scaling planewave results was obtained with increasing mesh. In both cases, chemical accuracy was readily attained. The efficiency of the approach for high temperature calculations was also shown, demonstrating decreasing cost with increasing temperature. The parallel scalability of the SQ method was demonstrated through strong and weak scaling simulations on upto more than hundred thousand processors involving tens of thousands of atoms. Comparison of the computational time with respect to a diagonalization based code at high temperature was presented that shows a two orders of magnitude speed-up of the SQ method over diagonalization for a 500 atom system. Overall, we conclude that the proposed method scales linearly, demonstrates the ability to simulate large systems of atoms and scale to a large number of processors.

6.1 Applications and Scope for future work

The SQ method is better suited for high temperature ab-initio molecular dynamics (AIMD) simulations [116]. The wall times taken by SQ in Section 5.5 indicate that it is capable of performing AIMD simulations for a few thousands of time steps involving few hundreds of atoms. Moreover, the perfect parallel scalability of SQ suggests that AIMD of even larger atomic systems with thousands of atoms is possible if sufficient

computational resources are available. On the other hand, diagonalization based codes are much slower and involve orthogonalization of many thousands of states and hence are not suitable for AIMD simulations at high temperature.

Another potential application for the SQ method is the study of mechanics of defects through multi-scale modeling and coarse graining techniques [84]. Since SQ is real space based and can perform infinite-cell calculations without recourse to k-point integration, different boundary conditions that are required to model defects can be easily incorporated into the simulation. Moreover, since the method is linear scaling and massively parallelizable, it is possible to simulate very large atomistic systems of the order of thousands of atoms that might be required to study the energetics of defects.

The development of the SQ method and the parallel code have been carried out under the Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx) at Lawrence Livermore National Laboratory (LLNL). We hope that further developments in the code will simultaneously help in developing next generation exascale architectures as well as in improving the performance of the code and the range of atomic systems accessible for simulations. The current supercomputers on which we tested the code have a peak performance on the order of petaflops i.e. 10^{15} FLOP/s. In the futuristic exascale systems, the performance is expected to cross 10^{18} FLOP/s. The current performance of the code is observed to be on the order of $10^{13} \sim 10^{14}$ FLOP/s. So, further optimizations and development at the machine level to improve the efficiency of the code are required to attain the maximum performance that can be offered by the processor architecture. Since the method scales linearly and has a very good parallel scaling behavior upto petascale level (due to minimal global communications), we expect that the code will scale from petascale to exascale level. We then hope that the DFT calculations using parallel SQ method can be a potential application that can make use of the exascale architectures very efficiently.

REFERENCES

[1] Alemany, M., Jain, M., Kronik, L., and Chelikowsky, J. R., "Real-space pseudopotential method for computing the electronic properties of periodic systems," *Physical Review B*, vol. 69, no. 7, p. 075101, 2004.

[2] Alfè, D., "Ab initio molecular dynamics, a simple algorithm for charge extrapolation," *Computer Physics Communications*, vol. 118, no. 1, pp. 31–33, 1999.

[3] Anantharaman, A. and Cancès, E., "Existence of minimizers for kohn–sham models in quantum chemistry," *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, vol. 26, no. 6, pp. 2425–2455, 2009.

[4] Anderson, D. G., "Iterative procedures for nonlinear integral equations," *Journal of the Association for Computing Machinery*, vol. 12, pp. 547–560, Oct. 1965.

[5] Anglade, P.-M. and Gonze, X., "Preconditioning of self-consistent-field cycles in density-functional theory: The extrapolar method," *Physical Review B*, vol. 78, no. 4, p. 045126, 2008.

[6] Artacho, E., Anglada, E., Diéguez, O., Gale, J. D., García, A., Junquera, J., Martin, R. M., Ordejón, P., Pruneda, J. M., Sánchez-Portal, D., and others, "The SIESTA method; developments and applicability," *Journal of Physics: Condensed Matter*, vol. 20, no. 6, p. 064208, 2008.

[7] Ashcroft, N. W. and Mermin, N. D., "Solid state phys," *Saunders, Philadelphia*, vol. 293, 1976.

[8] Banerjee, A. S., Suryanarayana, P., and Pask, J. E., "Periodic pulay method for robust and efficient convergence acceleration of self-consistent field iterations," *Chemical Physics Letters*, vol. 647, pp. 31 – 35, 2016.

[9] Bendt, P. and Zunger, A., "New approach for solving the density-functional self-consistent-field problem," *Physical Review B*, vol. 26, no. 6, p. 3114, 1982.

[10] Benzi, M., Boito, P., and Razouk, N., "Decay properties of spectral projectors with applications to electronic structure," *SIAM Review*, vol. 55, no. 1, pp. 3–64, 2013.

[11] Bobbitt, N. S., Schofield, G., Lena, C., and Chelikowsky, J. R., "High order forces and nonlocal operators in a Kohn-Sham Hamiltonian," *Phys. Chem. Chem. Phys.*, 2015. DOI: 10.1039/c5cp02561c.

[12] Bock, N., Challacombe, M., Gan, C. K., Henkelman, G., Nemeth, K., Niklasson, A. M. N., Odell, A., Schwegler, E., Tymczak, C. J., and Weber, V., "FreeON," 2014. Los Alamos National Laboratory (LA-CC 01-2; LA-CC-04-086), Copyright University of California.

[13] Bowler, D. R. and Miyazaki, T., "$\mathcal{O}(N)$ methods in electronic structure calculations," *Reports on Progress in Physics*, vol. 75, no. 3, p. 036503, 2012.

[14] Bowler, D. and Gillan, M., "An efficient and robust technique for achieving self consistency in electronic structure calculations," *Chemical Physics Letters*, vol. 325, no. 4, pp. 473–476, 2000.

[15] Brent, R. P., "An algorithm with guaranteed convergence for finding a zero of a function," *The Computer Journal*, vol. 14, no. 4, pp. 422–425, 1971.

[16] Briggs, W. L. and others, *The DFT: an owners' manual for the discrete Fourier transform*. Siam, 1995.

[17] Briggs, W. L., McCormick, S. F., and others, *A multigrid tutorial*. Siam, 2000.

[18] Broyden, C. G., "A class of methods for solving nonlinear simultaneous equations," *Mathematics of computation*, pp. 577–593, 1965.

[19] Cancès, E. and Le Bris, C., "Can we outperform the DIIS approach for electronic structure calculations?," *International Journal of Quantum Chemistry*, vol. 79, no. 2, pp. 82–90, 2000.

[20] Cancès, E. and Le Bris, C., "On the convergence of SCF algorithms for the Hartree-Fock equations," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 34, no. 04, pp. 749–774, 2000.

[21] Castro, A., Appel, H., Oliveira, M., Rozzi, C. A., Andrade, X., Lorenzen, F., Marques, M. A. L., Gross, E. K. U., and Rubio, A., "OCTOPUS: a tool for the application of time-dependent density functional theory," *Physica Status Solidi B-Basic Solid State Physics*, vol. 243, no. 11, pp. 2465–2488, 2006.

[22] Ceperley, D. M. and Alder, B. J., "Ground state of the electron gas by a stochastic method," *Phys. Rev. Lett.*, vol. 45, pp. 566–569, Aug 1980.

[23] Choly, N. and Kaxiras, E., "Kinetic energy density functionals for non-periodic systems," *Solid State Communications*, vol. 121, no. 5, pp. 281 – 286, 2002.

[24] Ciarlet, P., Lions, J., and Le Bris, C., *Handbook of Numerical Analysis : Special Volume: Computational Chemistry (Vol X)*. North-Holland, 2003.

[25] Courrieu, P., "Fast computation of Moore-Penrose inverse matrices," *arXiv preprint arXiv:0804.4809*, 2008.

[26] Dhillon, I. S., *A New O n2 Algorithm for the Symmetric Tridiagonal Eigenvalue Eigenvector Problem*. PhD thesis, University of California, Berkeley, 1997.

[27] Eyert, V., "A comparative study on methods for convergence acceleration of iterative vector sequences," *Journal of Computational Physics*, vol. 124, no. 2, pp. 271–285, 1996.

[28] Fang, H.-r. and Saad, Y., "Two classes of multisecant methods for nonlinear acceleration," *Numerical Linear Algebra with Applications*, vol. 16, no. 3, pp. 197–221, 2009.

[29] Frenkel, D. and Smit, B., *Understanding molecular simulation: from algorithms to applications*, vol. 1. Academic press, 2001.

[30] Ganine, V., Hills, N., and Lapworth, B., "Nonlinear acceleration of coupled fluid–structure transient thermal problems by Anderson mixing," *International Journal for Numerical Methods in Fluids*, vol. 71, no. 8, pp. 939–959, 2013.

[31] Garcia-Cervera, C. J., Lu, J., Xuan, Y., and E, W., "Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for Kohn-Sham density functional theory," *Physical Review B (Condensed Matter and Materials Physics)*, vol. 79, no. 11, p. 115110, 2009.

[32] Ghosh, S. and Suryanarayana, P., "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of Density Functional Theory. Part II: Periodic systems," *ArXiv e-prints*, Mar. 2016.

[33] Ghosh, S. and Suryanarayana, P., "Higher-order finite-difference formulation of periodic Orbital-free Density Functional Theory," *Journal of Computational Physics*, vol. 307, pp. 634–652, 2016.

[34] Ghosh, S. and Suryanarayana, P., "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of Density Functional Theory. Part I: Isolated clusters," *arXiv preprint arXiv:1603.04334*, 2016.

[35] Gillan, M. J., Bowler, D. R., Torralba, A. S., and Miyazaki, T., "Order-N first-principles calculations with the CONQUEST code," *Comput. Phys. Commun.*, vol. 177, pp. 14–18, JUL 2007.

[36] Goedecker, S., "Decay properties of the finite-temperature density matrix in metals," *Physical Review B*, vol. 58, no. 7, p. 3501, 1998.

[37] Goedecker, S., "Linear scaling electronic structure methods," *Rev. Mod. Phys.*, vol. 71, pp. 1085–1123, Jul 1999.

[38] Goedecker, S. and Colombo, L., "Efficient linear scaling algorithm for tight-binding molecular dynamics," *Physical Review Letters*, vol. 73, no. 1, p. 122, 1994.

[39] Goedecker, S. and Teter, M., "Tight-binding electronic-structure calculations and tight-binding molecular dynamics with localized orbitals," *Physical Review B*, vol. 51, no. 15, p. 9455, 1995.

[40] Gonze, X., Beuken, J. M., Caracas, R., Detraux, F., Fuchs, M., Rignanese, G. M., Sindic, L., Verstraete, M., Zerah, G., Jollet, F., Torrent, M., Roy, A., Mikami, M., Ghosez, P., Raty, J. Y., and Allan, D. C., "First-principles computation of material properties: the ABINIT software project," *Computational Materials Science*, vol. 25, pp. 478–492(15), 2002.

[41] Greenbaum, A., *Iterative methods for solving linear systems*, vol. 17. Siam, 1997.

[42] Gropp, W., Hoefler, T., Thakur, R., and Lusk, E., *Using Advanced MPI: Modern Features of the Message-Passing Interface*. MIT Press, 2014.

[43] Guide, M. U., "The mathworks," *Inc., Natick, MA*, vol. 5, 1998.

[44] Hackbusch, W., *Multi-grid methods and applications*, vol. 4. Springer Science & Business Media, 2013.

[45] Haydock, R., Heine, V., and Kelly, M., "Electronic-structure based on local atomic environment for tight-binding bands," *J Phys C Solid State*, vol. 5, pp. 2845–&, Jan 1972.

[46] Haydock, R., Heine, V., and Kelly, M., "Electronic-structure based on local atomic environment for tight-binding bands: Ii," *J Phys C Solid State*, vol. 8, pp. 2591–2605, Jan 1975.

[47] Hirose, K., Ono, T., Fujimoto, Y., and Tsukamoto, S., *First-principles calculations in real-space formalism: electronic configurations and transport properties of nanostructures*. World Scientific, 2005.

[48] Ho, K.-M., Ihm, J., and Joannopoulos, J., "Dielectric matrix scheme for fast convergence in self-consistent electronic-structure calculations," *Physical Review B*, vol. 25, no. 6, p. 4260, 1982.

[49] Hohenberg, P. and Kohn, W., "Inhomogeneous electron gas," *Physical Review*, vol. 136, no. 3B, pp. B864–B871, 1964.

[50] Jordan, D. and Mazziotti, D., "Spectral differences in real-space electronic structure calculations," *The Journal of Chemical Physics*, vol. 120, no. 2, pp. 574–578, 2003.

[51] Kerker, G., "Efficient iteration scheme for self-consistent pseudopotential calculations," *Physical Review B*, vol. 23, no. 6, p. 3082, 1981.

[52] Khabaza, I., "An iterative least-square method suitable for solving large sparse matrices," *The Computer Journal*, vol. 6, no. 2, pp. 202–206, 1963.

[53] Kleinman, L. and Bylander, D., "Efficacious form for model pseudopotentials," *Physical Review Letters*, vol. 48, no. 20, p. 1425, 1982.

[54] Kohn, W. and Sham, L. J., "Self-consistent equations including exchange and correlation effects," *Physical Review*, vol. 140, no. 4A, pp. A1133–A1138, 1965.

[55] Kresse, G. and Furthmüller, J., "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Physical Review B*, vol. 54, no. 16, p. 11169, 1996.

[56] Kresse, G. and Hafner, J., "Ab initio molecular dynamics for liquid metals," *Physical Review B*, vol. 47, no. 1, p. 558, 1993.

[57] Kronik, L., Makmal, A., Tiago, M. L., Alemany, M., Jain, M., Huang, X., Saad, Y., and Chelikowsky, J. R., "PARSEC–the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structures," *physica status solidi (b)*, vol. 243, no. 5, pp. 1063–1079, 2006.

[58] Kudin, K. N. and Scuseria, G. E., "Converging self-consistent field equations in quantum chemistry–recent achievements and remaining challenges," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 41, no. 02, pp. 281–296, 2007.

[59] Kudin, K. N., Scuseria, G. E., and Cances, E., "A black-box self-consistent field convergence algorithm: One step closer," *The Journal of Chemical Physics*, vol. 116, no. 19, pp. 8255–8261, 2002.

[60] Lanczos, C., *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[61] Langreth, D. C. and Mehl, M. J., "Beyond the local-density approximation in calculations of ground-state electronic properties," *Physical Review B*, vol. 28, no. 4, pp. 1809–1834, 1983.

[62] LeVeque, R., *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-Dependent Problems*. SIAM, 2007.

[63] Lin, L. and Yang, C., "Elliptic preconditioner for accelerating the self-consistent field iteration in kohn–sham density functional theory," *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. S277–S298, 2013.

[64] Lions, J.-L. and Ciarlet, P. G., *Handbook of Numerical Analysis: Computational chemistry*, vol. 10. Gulf Professional Publishing, 2003.

[65] Marks, L. and Luke, D., "Robust mixing for ab initio quantum mechanical calculations," *Physical Review B*, vol. 78, no. 7, p. 075114, 2008.

[66] Martin, R., *Electronic Structure: Basic theory and practical methods*. Cambridge University Press, 2004.

[67] Marx, D. and Hutter, J., *Ab initio molecular dynamics: basic theory and advanced methods*. Cambridge University Press, 2009.

[68] Mazziotti, D. A., "Spectral difference methods for solving differential equations," *Chemical Physics Letters*, vol. 299, no. 5, pp. 473–480, 1999.

[69] Mermin, N. D., "Thermal properties of the inhomogeneous electron gas," *Physical Review*, vol. 137, no. 5A, p. A1441, 1965.

[70] Mohr, S., Ratcliff, L. E., Boulanger, P., Genovese, L., Caliste, D., Deutsch, T., and Goedecker, S., "Daubechies wavelets for linear scaling density functional theory," *J. Chem. Phys.*, vol. 140, MAY 28 2014.

[71] Motamarri, P., Iyer, M., Knap, J., and Gavini, V., "Higher-order adaptive finite-element methods for orbital-free density functional theory," *Journal of Computational Physics*, vol. 231, no. 20, pp. 6596 – 6621, 2012.

[72] Niklasson, A. M., Steneteg, P., and Bock, N., "Extended Lagrangian free energy molecular dynamics," *The Journal of Chemical Physics*, vol. 135, no. 16, p. 164111, 2011.

[73] Ono, T. and Hirose, K., "Timesaving double-grid method for real-space electronic-structure calculations," *Phys. Rev. Lett.*, vol. 82, no. 25, pp. 5016–5019, 1999.

[74] Osei-Kuffuor, D. and Fattebert, J.-L., "Accurate and Scalable O(N) Algorithm for First-Principles Molecular-Dynamics Computations on Large Parallel Computers," *Phys. Rev. Lett.*, vol. 112, JAN 28 2014.

[75] Parr, R. and Yang, W., *Density-functional theory of atoms and molecules*. Oxford University Press, 1989.

[76] Pask, J. E., Klein, B. M., Fong, C. Y., and Sterne, P. A., "Real-space local polynomial basis for solid-state electronic-structure calculations: A finite-element approach," *Physical Review B*, vol. 59, no. 19, pp. 12352–12358, 1999.

[77] Pask, J. E. and Sterne, P. A., "Real-space formulation of the electrostatic potential and total energy of solids," *Phys. Rev. B*, vol. 71, p. 113101, Mar 2005.

[78] Pask, J. E., Sukumar, N., and Mousavi, S. E., "Linear scaling solution of the all-electron coulomb problem in solids," *International Journal for Multiscale Computational Engineering*, vol. 10, no. 1, pp. 83–99, 2012.

[79] Perdew, J. P. and Zunger, A., "Self-interaction correction to density-functional approximations for many-electron systems," *Phys. Rev. B*, vol. 23, pp. 5048–5079, May 1981.

[80] Perdew, J. P., Chevary, J. A., Vosko, S. H., Jackson, K. A., Pederson, M. R., Singh, D. J., and Fiolhais, C., "Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation," *Physical Review B*, vol. 46, no. 11, pp. 6671–6687, 1992.

[81] Perdew, J. P. and Wang, Y., "Accurate and simple analytic representation of the electron-gas correlation energy," *Physical Review B*, vol. 45, no. 23, p. 13244, 1992.

[82] Phillips, R., *Crystals, defects and microstructure: modeling across scales*. Cambridge University Press, 2001.

[83] Pickett, W. E., "Pseudopotential methods in condensed matter applications," *Computer Physics Reports*, vol. 9, no. 3, pp. 115–197, 1989.

[84] Ponga, M., Bhattacharya, K., and Ortiz, M., "A Sublinear-Scaling Approach to Density-Functional-Theory Analysis of Crystal Defects," *Journal of the Mechanics and Physics of Solids*, 2016.

[85] Potra, F. A. and Engler, H., "A characterization of the behavior of the Anderson acceleration on linear problems," *Linear Algebra and its Applications*, vol. 438, no. 3, pp. 1002–1011, 2013.

[86] Pratapa, P. P. and Suryanarayana, P., "Restarted Pulay mixing for efficient and robust acceleration of fixed-point iterations," *Chemical Physics Letters*, vol. 635, pp. 69 – 74, 2015.

[87] Pratapa, P. P. and Suryanarayana, P., "On numerically predicting the onset and mode of instability in atomistic systems," *Mechanics Research Communications*, 2016.

[88] Pratapa, P. P., Suryanarayana, P., and Pask, J. E., "Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems," *Journal of Computational Physics*, vol. 306, pp. 43–54, 2016.

[89] Pratapa, P. P., Suryanarayana, P., and Pask, J. E., "Spectral Quadrature method for accurate electronic structure calculations of metals and insulators," *Computer Physics Communications*, vol. 200, pp. 96 – 107, 2016.

[90] Prodan, E. and Kohn, W., "Nearsightedness of electronic matter," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 33, pp. 11635–11638, 2005.

[91] Pulay, P., "Convergence acceleration of iterative sequences. The case of SCF iteration," *Chemical Physics Letters*, vol. 73, no. 2, pp. 393–398, 1980.

[92] Quarteroni, A., Sacco, R., and Saleri, F., *Numerical mathematics*, vol. 37. Springer, 2007.

[93] Raczkowski, D., Canning, A., and Wang, L., "Thomas-Fermi charge mixing for obtaining self-consistency in density functional calculations," *Physical Review B*, vol. 64, no. 12, p. 121101, 2001.

[94] Rapaport, D. C., *The art of molecular dynamics simulation*. Cambridge university press, 2004.

[95] Rohwedder, T. and Schneider, R., "An analysis for the DIIS acceleration method used in quantum chemistry calculations," *Journal of mathematical chemistry*, vol. 49, no. 9, pp. 1889–1914, 2011.

[96] Ruiz-Serrano, A., Hine, N. D. M., and Skylaris, C.-K., "Pulay forces from localized orbitals optimized in situ using a psinc basis set," *J. Chem. Phys.*, vol. 136, JUN 21 2012.

[97] Saad, Y. and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[98] Saad, Y., *Iterative Methods for Sparse Linear System (2nd ed)*. SIAM, 2003.

[99] Shewchuk, J. R., "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[100] Skylaris, C.-K., Haynes, P. D., Mostofi, A. A., and Payne, M. C., "Introducing ONETEP: Linear-scaling density functional simulations on parallel computers," *The Journal of Chemical Physics*, vol. 122, no. 8, p. 084119, 2005.

[101] Soler, J. M., Artacho, E., Gale, J. D., García, A., Junquera, J., Ordejón, P., and Sánchez-Portal, D., "The SIESTA method for ab initio order-N materials simulation," *Journal of Physics: Condensed Matter*, vol. 14, no. 11, p. 2745, 2002.

[102] Srivastava, G., "Broyden's method for self-consistent field convergence acceleration," *Journal of Physics A: Mathematical and General*, vol. 17, no. 6, p. L317, 1984.

[103] Suryanarayana, P., "On spectral quadrature for linear-scaling Density Functional Theory," *Chemical Physics Letters*, vol. 584, pp. 182–187, 2013.

[104] Suryanarayana, P., Bhattacharya, K., and Ortiz, M., "A mesh-free convex approximation scheme for Kohn-Sham density functional theory," *Journal of Computational Physics*, vol. 230, no. 13, pp. 5226 – 5238, 2011.

[105] Suryanarayana, P., Bhattacharya, K., and Ortiz, M., "Coarse-graining Kohn-Sham Density Functional Theory," *Journal of the Mechanics and Physics of Solids*, vol. 61, no. 1, pp. 38 – 60, 2013.

[106] Suryanarayana, P., Gavini, V., Blesgen, T., Bhattacharya, K., and Ortiz, M., "Non-periodic finite-element formulation of Kohn-Sham density functional theory," *Journal of the Mechanics and Physics of Solids*, vol. 58, no. 2, pp. 256 – 280, 2010.

[107] Suryanarayana, P. and Phanish, D., "Augmented Lagrangian formulation of orbital-free density functional theory," *Journal of Computational Physics*, vol. 275, pp. 524 – 538, 2014.

[108] Suryanarayana, P., Pratapa, P. P., and Pask, J. E., "Alternating Anderson-Richardson method: An efficient alternative to preconditioned Krylov methods for large, sparse linear systems," *arXiv preprint arXiv:1606.08740*, 2016.

[109] Tadmor, E. B. and Miller, R. E., *Modeling materials: continuum, atomistic and multiscale techniques*. Cambridge University Press, 2011.

[110] Trefethen, L. N., "Is Gauss quadrature better than Clenshaw-Curtis?," *SIAM review*, vol. 50, no. 1, pp. 67–87, 2008.

[111] Troullier, N. and Martins, J. L., "Efficient pseudopotentials for plane-wave calculations," *Physical Review B*, vol. 43, no. 3, p. 1993, 1991.

[112] Tsuchida, E., "Augmented orbital minimization method for linear scaling electronic structure calculations," *J. Phys. Soc. Jpn.*, vol. 76, MAR 2007.

[113] Vanderbilt, D. and Louie, S. G., "Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method," *Physical Review B*, vol. 30, no. 10, p. 6118, 1984.

[114] Walker, H. F. and Ni, P., "Anderson acceleration for fixed-point iterations," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715–1735, 2011.

[115] Weinan, E., *Principles of multiscale modeling*. Cambridge University Press, 2011.

[116] White, T., Richardson, S., Crowley, B., Pattison, L., Harris, J., and Gregori, G., "Orbital-free density-functional theory simulations of the dynamic structure factor of warm dense aluminum," *Physical Review Letters*, vol. 111, no. 17, p. 175002, 2013.

[117] Willert, J., Taitano, W. T., and Knoll, D., "Leveraging Anderson Acceleration for improved convergence of iterative solutions to transport systems," *Journal of Computational Physics*, vol. 273, pp. 278–286, 2014.

[118] Yang, X. I. and Mittal, R., "Acceleration of the Jacobi iterative method by factors exceeding 100 using scheduled relaxation," *Journal of Computational Physics*, vol. 274, pp. 695 – 708, 2014.